



Universidad Carlos III de Madrid

Escuela Politécnica Superior

TRABAJO FIN DE GRADO

# Sistema de navegación y gestión de rutas para vehículos eléctricos en Android

Francisco Vara De la Cruz – Doblado

GRADO EN INGENIERÍA DE SISTEMAS AUDIOVISUALES

Tutor: Israel González Carrasco

Septiembre 2015



## AGRADECIMIENTOS

Es difícil echar la vista atrás y buscar gente a quien dar las gracias, ya que en mi caso no sólo es un trabajo de los años que ha durado la carrera sino el de toda una vida hasta llegar a donde he llegado. Habrá personas que se quedarán fuera de esta mención, pero no por ello son menos importantes.

En primer lugar, he de agradecer a mi padre, Vicente, el apoyo y ánimo transmitido, ya que sin su ayuda y sus consejos nada hubiera sido posible. Desde el primer momento creyó en mí y sabía que con esfuerzo y ayuda lo conseguiría. Gracias.

Muchas gracias a mi madre, Encarnación, mis hermanos, Vicente y Carlos, y a mi familia, porque, consciente o inconscientemente, me han ayudado y han tenido paciencia conmigo en los momentos más duros. Su apoyo ha facilitado el seguir con esto día a día.

Gracias, a mis actuales compañeros de Unisys por enseñarme todo lo aprendido sobre dispositivos móviles, a Luis Galey por darme la oportunidad de entrar en su departamento, y en especial a Jose Ángel Maneiro ya que sin sus horas dedicadas en consejos y enseñanzas sobre Android este proyecto no habría visto la luz.

Por último, agradecer a mi tutor Israel la dedicación y paciencia que ha tenido a lo largo del proyecto, además de su buena disposición en todo momento.



## RESUMEN

El vehículo es un medio de transporte necesario para la sociedad actual, siendo una de las mayores fuentes de contaminación en las ciudades causada por las emisiones procedentes de la gran mayoría de los vehículos que están actualmente en circulación. Estos vehículos contribuyen al efecto invernadero y al agotamiento de materias primas no renovables, por ello aparecen nuevos conceptos de medios de transportes para su reducción. Son muchos planes de acción los que se están llevando a cabo en todo el mundo para erradicar estas emisiones, pero la gran mayoría los desconoce.

Por otra parte, la entrada al mercado de Smartphones en la población es bastante alta, proporcionándonos cada día nuevas funcionalidades y más facilidades para estar más informados. Android se ha impuesto como uno de los principales sistemas operativos de la actual generación de dispositivos móviles.

Este proyecto ha juntado estos dos conceptos desarrollando una aplicación que ayude al usuario a concienciarse del uso de los vehículos eléctricos, aportando al usuario una herramienta para la gestión de los mismos, mostrando que cada vez presentan una autonomía mayor para realizar una ruta. Además proporciona al alumno con el refuerzo y ampliación de los conocimientos adquiridos al respecto durante los años de carrera.

A lo largo de este documento se dan a conocer todas estas tecnologías, describiendo el diseño y desarrollo de la aplicación.

**Palabras clave:** aplicación, Android, vehículo eléctrico, mapas, Movele



## ABSTRACT

Vehicles are a necessary mean of transport for our current society. Nevertheless, it is considered to be one of the main sources of pollution due to the large quantity of eco-unfriendly vehicles in use. These vehicles contribute to the greenhouse effect and the depletion of the non-renewable raw materials. In order to reduce these negative consequences, new concepts of means of transports arise. There are several projects on the reduction of pollution that are operating all around the world. However, most of people are unaware of these projects.

On the other hand, the higher entrance of Smartphones in the market provides us with new functionalities and facilities to keep daily informed. Android has managed to impose as one of the main operating systems in the current generation of mobile devices.

This project has worked on both concepts: developing an application which will make users aware of the utility in the use of electronic vehicles and offering a tool to manage the increasing autonomy of their electric vehicles when they have to perform a route. According to its author, this project has provided the opportunity to deeply investigate on a field which was previously introduced in the degree. Far from the acquired basic concepts, this project has achieved greater expectations.

**Key words:** application, Android, electric vehicle, maps, Movele



## ÍNDICE GENERAL

AGRADECIMIENTOS.....	1
RESUMEN .....	2
ABSTRACT .....	3
ÍNDICE GENERAL.....	4
ÍNDICE DE FIGURAS .....	7
ÍNDICE DE TABLAS .....	10
Capítulo 1: Introducción .....	13
1.1. Motivación .....	13
1.2. Objetivos .....	14
1.3. Estructura de la memoria .....	14
Capítulo 2. Planteamiento del problema. ....	16
2.1. Sistemas operativos móviles.....	16
2.1.2. Android.....	18
2.1.2. iOS .....	19
2.1.3 Windows Phone .....	19
2.1.4. Symbian OS .....	20
2.1.5. Ubuntu Phone .....	21
2.2. Android .....	21
2.2.1. Pila software de Android.....	22
2.2.2. Arquitectura .....	23
2.2.3. Entorno de desarrollo .....	24
2.2.4. Versiones.....	25
2.3. Google.....	26
2.3.1. Google Developers .....	29
2.3.2. Google Maps .....	29
2.3.3. Google Play.....	29
2.4. Backend as a service .....	30
2.5. Vehículo eléctrico .....	31



2.6. Movele .....	34
Capítulo 3. Análisis de los sistemas .....	36
3.1. Requisitos.....	36
3.1.1. Requisitos Comunes De Los Interfaces .....	36
3.1.2. Requisitos funcionales.....	37
3.1.3. Requisitos no funcionales .....	40
3.2. Casos de uso .....	43
3.3. Análisis de alternativas de la solución .....	47
3.3.1. Alternativas del servicio de puntos de recarga .....	47
3.3.2. Alternativas de la base de datos .....	48
Capítulo 4. Diseño de la solución técnica .....	49
4.1. Base de datos.....	49
4.2. Google Maps API.....	51
4.2.1. Google Play Services.....	51
4.2.2. API Key de Google .....	52
4.2.3. Servicio de rutas .....	56
4.3. Parsear página web.....	59
4.4. Control de versiones .....	60
4.5. Formulación .....	61
4.6. Versión mínima .....	62
4.7. Interfaz de usuario .....	63
4.7.1. Estudio de diseño .....	63
4.7.2. Pantalla de inicio y registro .....	64
4.7.3. Pantalla principal.....	66
4.7.4. Pantallas del garaje y del concesionario .....	68
4.7.5. Iconos .....	69
Capítulo 5. Pruebas de los sistemas y manual de usuario .....	72
5.1. Pruebas de los sistemas .....	72
5.2. Manual de usuario .....	77
Capítulo 6. Planificación del trabajo y presupuesto .....	87



6.1. Planificación del trabajo .....	87
6.2. Presupuesto .....	89
Capítulo 7. Conclusiones y líneas futuras .....	92
7.1. Conclusiones .....	92
7.2. Líneas futuras.....	92
Bibliografía .....	94
Chapter 1: Introduction .....	95
1.1. Motivation .....	95
1.2. Objectives .....	96
1.3. Structure of the report .....	96
Chapter 7: Conclusions and future lines of research.....	97
7.1. Conclusions .....	97
7.2. Future lines of research .....	97
SUMMARY .....	99



## ÍNDICE DE FIGURAS

Figura 1: Sistemas operativos móviles más utilizados en el mundo.....	17
Figura 2: Sistemas operativos móviles más utilizados en Europa .....	18
Figura 3: Logotipo del sistema operativo Android .....	19
Figura 4: Logotipo del sistema operativo iOS .....	19
Figura 5: Logotipo del sistema operativo Windows Phone .....	20
Figura 6: Logotipo del sistema operativo Windows 10 .....	20
Figura 7: Logotipo del sistema operativo Symbian.....	20
Figura 8: Logotipo del sistema operativo Ubuntu .....	21
Figura 9: Primer teléfono en el mercado con el sistema Android .....	21
Figura 10: Pila software del sistema operativo Android.....	23
Figura 11: Primer logotipo de la compañía Google .....	26
Figura 12: Motores de búsqueda más utilizados a nivel mundial .....	27
Figura 13: Logotipo de Google en 2015.....	28
Figura 14: Vehículo eléctrico del siglo XIX .....	32
Figura 15: Logotipo del Plan Movele .....	34
Figura 16: Punto de recarga Movele .....	35
Figura 17: Representación esquemática de los casos de uso.....	43
Figura 18: Relación entre tablas .....	49
Figura 19: Tabla "Users" de Parse .....	50
Figura 20: Tabla "Cars" de Parse .....	50
Figura 21: Tabla "Garage" de Parse.....	50
Figura 22: SDK Manager de Android .....	52
Figura 23: Creación de un proyecto nuevo.....	53
Figura 24: Consola de desarrolladores de Google .....	54
Figura 25: Comandos clave debug en un terminal iOS .....	55





Figura 26: Parte de código del archivo AndroidManifest.xml .....	55
Figura 27: Comandos clave producción en un terminal iOS .....	56
Figura 28: Pantalla de inicio de sesión .....	64
Figura 29: Mensaje de error .....	64
Figura 30: Pantalla de registro de usuario .....	65
Figura 31: Pantalla de inicio de sesión con teclado .....	66
Figura 32: Pantalla principal .....	67
Figura 33: Pantallas de Garaje y Concesionario .....	68
Figura 34: Icono principal de la aplicación .....	69
Figura 35: Cuadrícula para iconos de Google .....	70
Figura 36: Plantilla iconos .....	70
Figura 37: Vista de acciones creadas .....	71
Figura 38: Iconos del mapa .....	71
Figura 39: Icono temporal de vehículo .....	71
Figura 40: Dispositivos utilizados .....	72
Figura 41: Código QR para descargar la aplicación .....	76
Figura 42: Pantallas de inicio de sesión y registro de usuario .....	77
Figura 43: Pantalla principal .....	78
Figura 44: Botón Garaje .....	78
Figura 45: Sección Garaje .....	79
Figura 46: Sección Concesionario .....	80
Figura 47: Sección Garaje con un vehículo seleccionado .....	81
Figura 48: Pantalla principal con una ruta indicada .....	82
Figura 49: Puntos de gasto de batería y punto de recarga cercano .....	83
Figura 50: Menú de la aplicación .....	83
Figura 51: Opciones de la ruta .....	84
Figura 52: Sección Acerca de .....	84



Figura 53: Botón puntos de recarga .....	85
Figura 54: Puntos de recarga de la posición del usuario .....	85
Figura 55: Opciones de Google Maps.....	86
Figura 56: Main screen .....	103



## ÍNDICE DE TABLAS

Tabla 1: Plantilla requisito funcional .....	37
Tabla 2: Requisito funcional (I) .....	38
Tabla 3: Requisito funcional (II) .....	38
Tabla 4: Requisito funcional (III) .....	38
Tabla 5: Requisito funcional (IV) .....	38
Tabla 6: Requisito funcional (V) .....	38
Tabla 7: Requisito funcional (VI) .....	39
Tabla 8: Requisito funcional (VII) .....	39
Tabla 9: Requisito funcional (VIII) .....	39
Tabla 10: Requisito funcional (IX) .....	39
Tabla 11: Requisito no funcional (I) .....	40
Tabla 12: Requisito no funcional (II) .....	40
Tabla 13: Requisito no funcional (III) .....	40
Tabla 14: Requisito no funcional (IV) .....	41
Tabla 15: Requisito no funcional (V) .....	41
Tabla 16: Requisito no funcional (VI) .....	41
Tabla 17: Requisito no funcional (VII) .....	41
Tabla 18: Requisito no funcional (VIII) .....	42
Tabla 19: Requisito no funcional (IX) .....	42
Tabla 20: Requisito no funcional (X) .....	42
Tabla 21: Requisito no funcional (XI) .....	42
Tabla 22: Caso de uso (I) .....	43
Tabla 23: Caso de uso (II) .....	44
Tabla 24: Caso de uso (III) .....	44
Tabla 25: Caso de uso (IV) .....	44



Tabla 26: Caso de uso (V) .....	44
Tabla 27: Caso de uso (VI) .....	44
Tabla 28: Caso de uso (VII) .....	45
Tabla 29: Caso de uso (VIII) .....	45
Tabla 30: Caso de uso (IX).....	45
Tabla 31: Caso de uso (X).....	45
Tabla 32: Caso de uso (XI).....	45
Tabla 33: Caso de uso (XII).....	46
Tabla 34: Plantilla de prueba realizada .....	73
Tabla 35: Prueba realizada (I) .....	73
Tabla 36: Prueba realizada (II) .....	73
Tabla 37: Prueba realizada (III) .....	73
Tabla 38: Prueba realizada (IV).....	73
Tabla 39: Prueba realizada (V).....	74
Tabla 40: Prueba realizada (VI).....	74
Tabla 41: Prueba realizada (VII).....	74
Tabla 42: Prueba realizada (VIII).....	74
Tabla 43: Prueba realizada (IX) .....	74
Tabla 44: Prueba realizada (X) .....	74
Tabla 45: Prueba realizada (XI) .....	75
Tabla 46: Prueba realizada (XII) .....	75
Tabla 47: Prueba realizada (XIII) .....	75
Tabla 48: Prueba realizada (XIV).....	75
Tabla 49: Prueba realizada (XV).....	75
Tabla 50: Prueba realizada (XVI).....	75
Tabla 51: Diagrama de Gantt (I) .....	88
Tabla 52: Diagrama de Gantt (II) .....	89



Tabla 53: Coste del proyecto .....	90
Tabla 54: Margen de beneficio.....	91
Tabla 55: Margen de riesgo .....	91
Tabla 56: Cantidad total .....	91



## Capítulo 1: Introducción

### 1.1. Motivación

El vehículo es un medio de transporte necesario para las sociedades tal y como las conocemos hoy en día, pero también es una de las mayores fuentes de contaminación en las ciudades. La contaminación por el tráfico es causada por la quema de combustibles fósiles, como el gasoil y la gasolina procedentes de vehículos de combustión interna, los más utilizados en el presente.

Las emisiones de la mayoría de vehículos que están en circulamiento causan un gran problema medioambiental contribuyendo al efecto invernadero y al agotamiento de materias primas no renovables, por ello aparecen nuevos conceptos de medios de transportes para su reducción.

La comisión Europea ha propuesto un plan Global para erradicar el vehículo tradicional para el año 2050. Son muchos planes de acción los que se están implantando en todo el mundo, pero la gran mayoría los desconoce.

Por otra parte, los dispositivos móviles están evolucionando a pasos enormes, ofreciendo a los usuarios una mayor comunicación, siendo una herramienta fundamental tanto en el trabajo como para el entretenimiento. Los conocimientos adquiridos en la carrera han servido como introducción a este nuevo mundo en el que este proyecto se sumerge de lleno. Por ello se decidió escoger un proyecto en el que se profundizara en la materia para comprender las complejidades de su funcionamiento.

Estos hechos suponen una gran motivación en la realización de este proyecto ya que además de proporcionar al alumno una mejora en sus aptitudes sobre este campo, descubre que la tecnología puede ayudar a terminar con el desconocimiento de todos estos planes de acción que pueden paliar los graves problemas medioambientales a los que nos enfrentamos.



## 1.2. Objetivos

En el presente trabajo fin de grado se desea lograr dos objetivos. El primero, es el desarrollo de una aplicación Android orientada a mapas. Esta deberá proporcionar al consumidor una buena experiencia de usuario, aportando información útil y un fácil manejo.

La interfaz gráfica deberá ser sencilla y a la vez tener un aspecto acorde con las actuales tendencias. Estos puntos no siempre se llevan a cabo por los desarrolladores, teniendo un equipo de diseño que les aconsejan. Por ello, se deberá llevar a cabo tareas tanto de desarrollo como de diseño.

El segundo objetivo es la creación de un proyecto capaz de ayudar al usuario a una concienciación del uso de vehículos eléctricos, enseñando el plan de ayudas Movele del Ministerio y mostrando la autonomía cada vez mayor de los mismos para realizar una ruta.

La aplicación debe ser capaz de implementar las siguientes funcionalidades:

- Manejo de mapas de Google Maps.
- Cálculo de autonomía según la ruta indicada.
- Mostrar puntos de recarga de vehículos eléctricos cerca del punto donde el usuario consumiera la totalidad de su batería.
- Indicar los puntos de recarga cercanos al punto actual de localización del usuario.

## 1.3. Estructura de la memoria

La presente memoria del trabajo de fin de grado se desglosa en 7 capítulos, los cuales se describen a continuación:

- Capítulo 1. Se recoge la motivación del proyecto y los objetivos que se pretenden alcanzar, incluyendo la estructura de la memoria.
- Capítulo 2. Presenta el planteamiento del problema, exponiendo un análisis del mismo, determinando las técnicas y tecnologías que se han tenido en cuenta para el diseño del problema.
- Capítulo 3. Se hace referencia, en forma de requisitos, a las capacidades, funcionalidades y restricciones de los sistemas, analizando las diferentes alternativas surgidas.
- Capítulo 4. Se expone la solución técnica seleccionada, explicando el método llevado a cabo para el desarrollo de la aplicación.



- Capítulo 5. Se recogen las pruebas realizadas para la presentación de la versión definitiva de la aplicación. Además, se proporciona un manual de usuario, en el cual se muestran las opciones que dispone.
- Capítulo 6. Se presenta la planificación y el presupuesto económico para la realización del proyecto.
- Capítulo 7. Se exponen las conclusiones finales del proyecto, añadiendo las posibles líneas futuras de mejora de la aplicación.





## Capítulo 2. Planteamiento del problema.

En este capítulo se recogen las bases teóricas que engloban el proyecto, resumiendo las tecnologías y servicios que han sido utilizados. Se expondrá un resumen de los servicios en los que el proyecto está involucrado y se describirá con detalle el formato y temática de la aplicación desarrollada.

### 2.1. Sistemas operativos móviles.

Un sistema operativo (Operating System, OS) se puede definir como un software que, tras ser cargado inicialmente en un equipo, hace que funcione, proporcionándole un entorno para la ejecución de programas. Estos programas, que suelen llamarse aplicaciones o programas de aplicación, hacen uso del sistema operativo como intermediario para acceder a los recursos del sistema. Cuando citamos recursos del sistema nos referimos al procesador, archivos o los dispositivos de entrada y salida, que actúan gestionando la memoria, respondiendo a las peticiones del usuario y optimizando el rendimiento del equipo. Por ello, un sistema operativo será la base sobre la que se instalarán los programas de aplicación.

Los dispositivos móviles, que actualmente conocemos como Smartphones (teléfonos inteligentes), han evolucionado de tal manera que incluyen sistemas operativos con características similares a las de un ordenador. Su sistema operativo está enfocado en la movilidad, la conexión inalámbrica y la administración del procesamiento, del almacenamiento y del consumo de energía.

Los Smartphones fueron diseñados para ofrecer a los usuarios una mayor comunicación y transmisión de datos, lo cual se ha conseguido siendo una herramienta fundamental tanto para el trabajo como para el entretenimiento.

Normalmente los sistemas operativos móviles se basan en el modelo de capas, contando con unas capas específicas, aunque pueden cambiar según el funcionamiento del equipo. A continuación se detallan estas capas:

- **Kernel.** El núcleo o Kernel, es la capa que permite el acceso a los diferentes elementos hardware del equipo. Proporciona los drivers (controladores), la gestión de procesos y el acceso a la gestión de memoria a las capas superiores. Algunos de los núcleos utilizados en los dispositivos móviles son Linux para Android, XNU para iOS o kernels propios como es el caso del núcleo de Windows Mobile.

- **Middleware.** La capa Middleware o intermediador es la encargada de que las aplicaciones y librerías, entre otras, puedan ser ejecutadas. Ofrece servicios transparentes para el usuario, ya que él no debe configurar nada para que se ejecuten correctamente. Algunos de estos servicios son el motor de comunicaciones y mensajería, seguridad, gestión móvil o codecs (codificador – decodificador) multimedia.
- **Entorno de ejecución de aplicaciones.** Facilita el conjunto de herramientas necesarias a un desarrollador para la creación de programas para el sistema operativo en el que programe. En esta capa encontramos el gestor de aplicaciones y un conjunto de APIs (Interfaz de Programación de Aplicaciones).
- **Interfaz de usuario.** La interfaz de usuario es la utilizada para interactuar con el equipo. Incluye los componentes gráficos necesarios para poder utilizar el equipo, tales como botones, menús, pantallas, etc.

En los siguientes gráficos podemos comparar los usos de los principales sistemas operativos móviles.

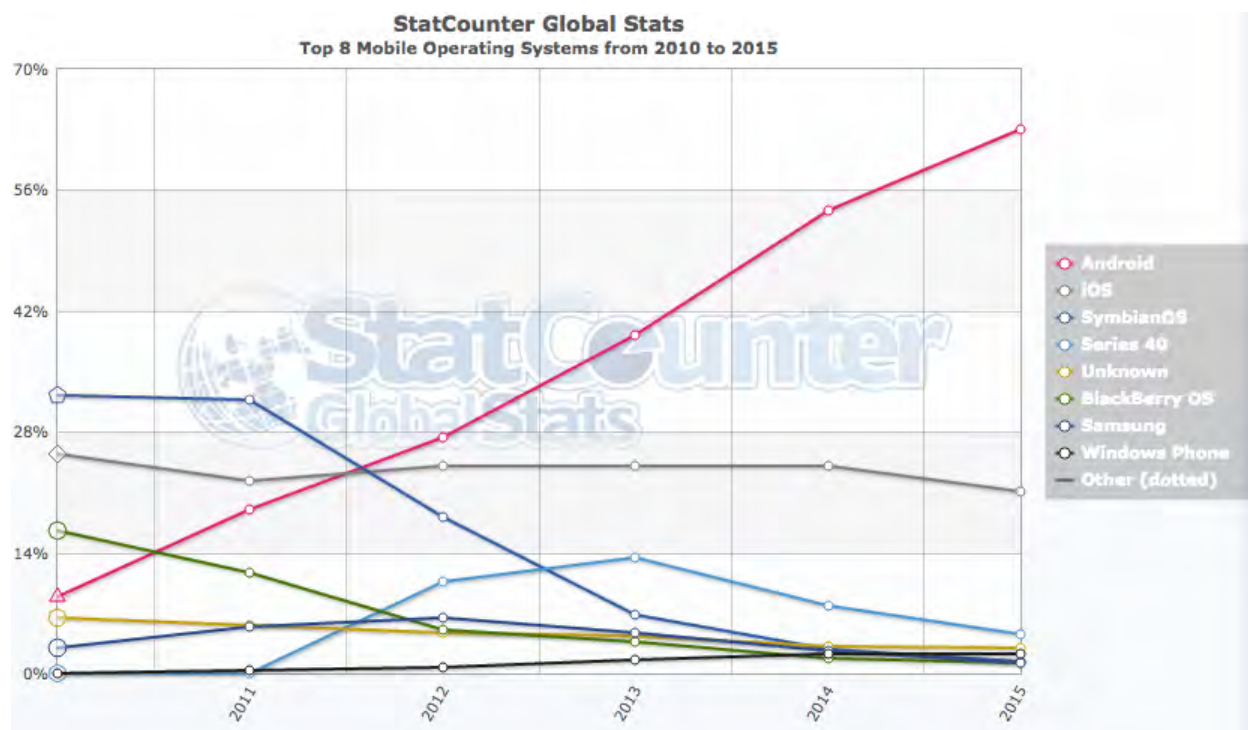


Figura 1: Sistemas operativos móviles más utilizados en el mundo

Como podemos observar en los últimos años Android e iOS ha superado al sistema operativo Symbian que era el más usado en todo el mundo hace unos años. Estos datos son globales por lo que hay que tener en cuenta que según el continente van a utilizarse más unos u otros, ya que el poder adquisitivo y el uso de las tecnologías varían.

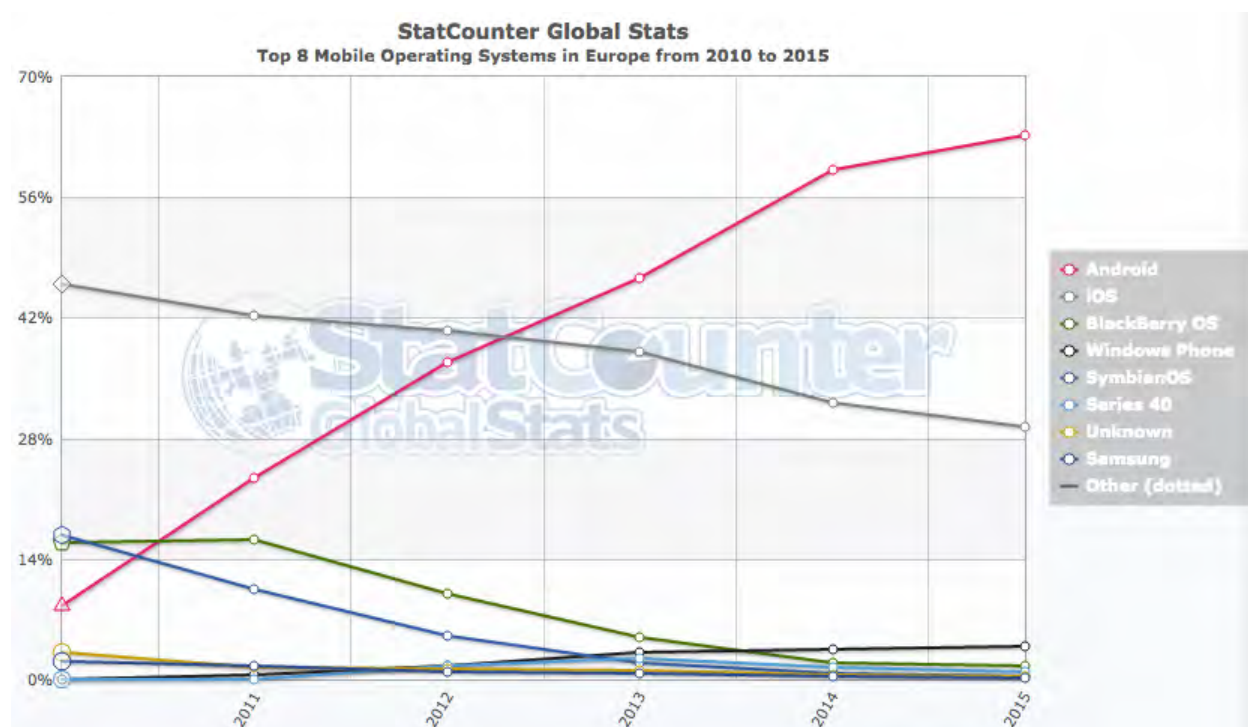


Figura 2: Sistemas operativos móviles más utilizados en Europa

En Europa se observa que actualmente los sistemas más usados, con diferencia, son Android e iOS, ya que en los últimos años Android ha generado un mayor crecimiento.

### 2.1.2. Android.

El sistema operativo Android fue originalmente diseñado para dispositivos móviles como Smartphones (teléfonos inteligentes) y tablets, pero al ser comprado por Google en el 2005 fue modificado para ser utilizado en una gran variedad, cada vez más amplia, de dispositivos.

Android está basado en una versión modificada del kernel de Linux convirtiendo el código escrito de Java de las aplicaciones a código capaz de comprender por el núcleo, a través de su entorno de ejecución. Es un sistema de código abierto, por lo que cualquier persona puede crear y desarrollar aplicaciones para el sistema.



Figura 3: Logotipo del sistema operativo Android

### 2.1.2. iOS

Sistema operativo creado por Apple originalmente para dispositivos iPhone, pero más tarde ampliado a otros productos de la marca. Está basado en Darwin con un núcleo XNU.

Una de sus grandes características es su propiedad exclusiva por Apple, no puede ser usado por otras compañías, siendo necesaria la adquisición de un dispositivo de la compañía para poder utilizarlo. Esto hace que sus ventas sean menores en muchos países, comparándolas con el sistema operativo de Android, pero también trae sus beneficios ya lo convierte en un sistema más robusto y seguro al siempre estar respaldado por una sola compañía.



Figura 4: Logotipo del sistema operativo iOS

### 2.1.3 Windows Phone

Windows Phone es un sistema móvil desarrollado por Microsoft enfocado al mercado de consumo, y no como su predecesor que estaba centrado más al mercado empresarial. Este sistema dispone de un núcleo centrado en el sistema operativo Windows NT.

Su sucesor está actualmente en pruebas, el sistema Windows 10 Mobile. Esta última versión es una variante del sistema operativo Windows 10, la cual tendrán una fácil integración en cualquier tipo de plataforma.



Figura 5: Logotipo del sistema operativo Windows Phone



Figura 6: Logotipo del sistema operativo Windows 10

#### 2.1.4. Symbian OS

El sistema operativo Symbian fue creado por varias compañías de telefonía móvil entre las que se encuentran, Sony Ericsson, PSION, Samsung, Siemens, Benq, LG, Motorola o Nokia, la cual es la actual propietaria del sistema. El objetivo para el cual fue creado fue para competir con las PDAs (Personal Digital Assistant, agenda electrónica de bolsillo) de la época, una competición que actualmente a evolucionado a los dispositivos móviles como Android e iOS.

Su mayor ventaja es que sus aplicaciones están orientadas al manejo de eventos, ya que el uso de múltiples hilos, como hacen algunos de sus competidores, producen sobrecarga en el sistema, lo que le convierte en un sistema operativo muy fiable.



Figura 7: Logotipo del sistema operativo Symbian

### 2.1.5. Ubuntu Phone

Ubuntu Phone está basado en Linux y desde su creación en 2013 está siendo bien acogido por los consumidores, apostando por este sistema operativo marcas como Meizu y Bq. Cuenta con las tecnologías que utiliza su predecesor en ordenadores Ubuntu, por lo que presenta compatibilidad con sus aplicaciones de escritorio.



Figura 8: Logotipo del sistema operativo Ubuntu

### 2.2. Android

Sistema operativo basado en el kernel de Linux, proyecto de la Open Handset Alliance (OHA). El Open Handset Alliance es un consorcio de 84 compañías de hardware, software y telecomunicaciones que desarrollan estándares abiertos para el sector de la telefonía móvil. Algunos de sus miembros más destacados son Google, Samsung, Intel, Dell, Nvidia, etc.

Bajo una licencia de código abierto, el lanzamiento de Android tenía la intención de competir contra otras plataformas móviles como Apple, Symbian, Nokia o Microsoft. El primer dispositivo lanzado al mercado con la plataforma de Android fue el HTC Dream (T-Mobile G1) a finales del 2008.



Figura 9: Primer teléfono en el mercado con el sistema Android



Los desarrolladores son los responsables de sacar todo el partido a las plataformas del mercado, por ello, Android les brindó con todas las herramientas para conseguirlo. Les proporcionó el Android SDK (Software Development Kit), que dispone de las herramientas y APIs necesarias para iniciar a un nuevo usuario en la programación de Android. Además, para competir con su rival más fuerte, Apple, desarrollar en Android es posible desde cualquier sistema operativo y desde varios IDEs (Entorno de desarrollo interactivo), no como Apple que sólo permite su uso desde su sistema operativo OS.

### 2.2.1. Pila software de Android

La arquitectura de Android se resume en un núcleo Linux y una colección de librerías escritas en lenguaje C/C++, aunque para su programación es necesario utilizar el lenguaje Java para la parte de aplicación y un editor propio de Android o lenguaje XML para la parte gráfica.

- **Kernel de Linux.** El núcleo del sistema operativo Android es una versión 2.6 de Linux, similar al de cualquier equipo Linux. Dota al sistema con una capa de abstracción entre el hardware y el resto de la pila de su arquitectura, permitiéndonos acceder a los servicios básicos como los controladores hardware, procesos y gestión de la memoria, seguridad, la red y la administración de la energía.
- **Librerías.** En la siguiente capa se encuentran la librerías nativas de Android. Proporciona librerías como C/C++ estándar, reproducción de audio/video, gestión de pantalla, bibliotecas gráficas como SGL y OpenGL para 2D y 3D, SQLite para el soporte de bases de datos nativas, WebKit y SSL para el navegador web integrado y la seguridad en internet, etc.
- **Entorno de ejecución.** Puede formar parte de las librerías, ya que se compone de las bibliotecas del núcleo (Java y Android) y la máquina virtual Dalvik. Esta máquina virtual ejecuta las aplicaciones no nativas de Android, compilándolas en un formato específico para asegurarnos que se podrán ejecutar varias instancias en cualquier dispositivo de manera eficiente.
- **Framework de aplicación** (Marco de aplicación). Formada por las clases para crear aplicaciones Android, permitiendo el acceso al hardware y gestionando la interfaz de aplicaciones y recursos del usuario.
- **Nivel de aplicación.** Donde se encuentran las aplicaciones, tanto nativas o de desarrolladores. Por ello se tiene un control total por el usuario a todo el software, ya que pueden acceder a los mismos servicios que ofrece el sistema operativo.



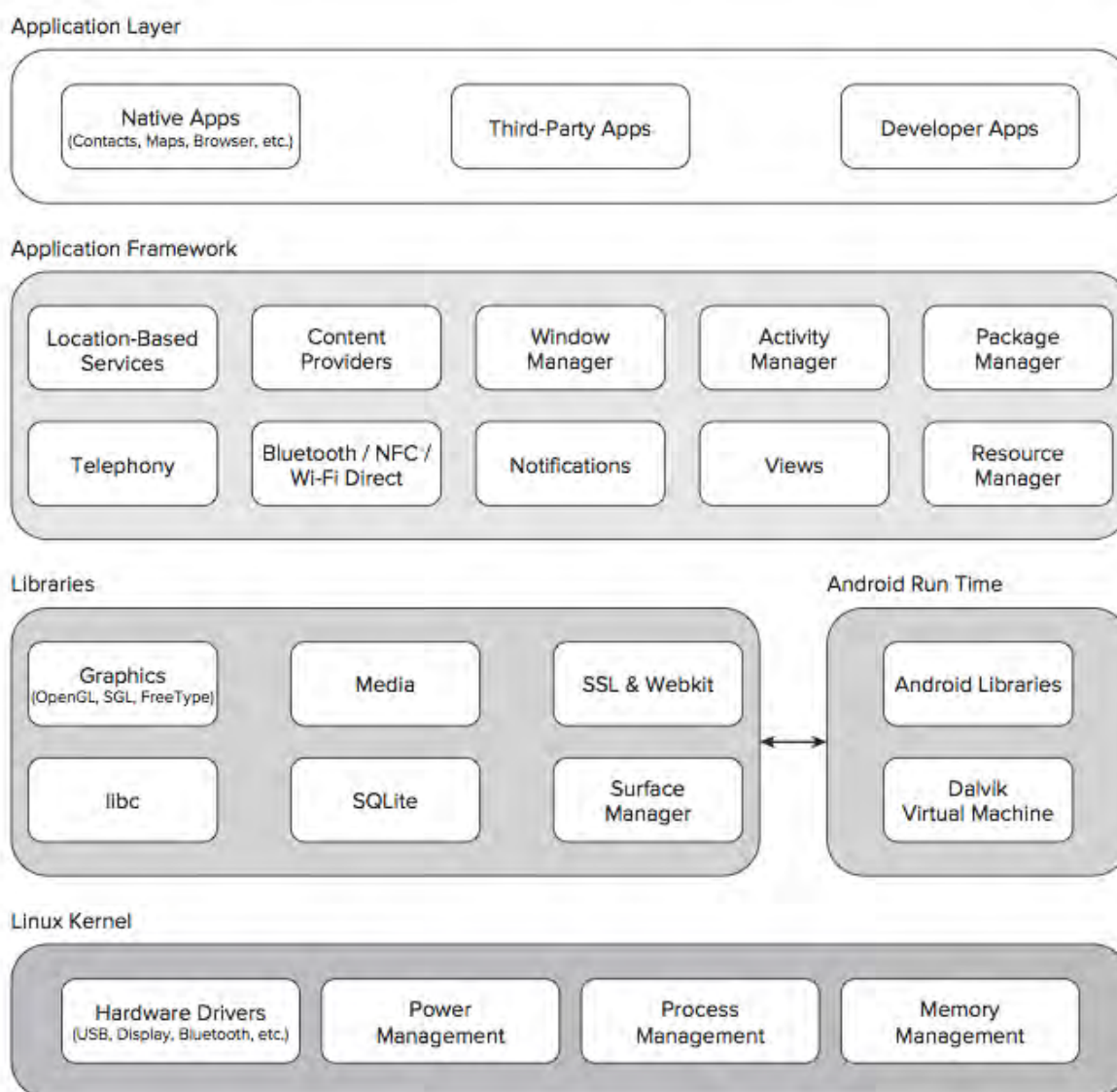


Figura 10: Pila software del sistema operativo Android

### 2.2.2. Arquitectura

La estructura del sistema Android se basa en componentes. Una aplicación será siempre una combinación de unos o más de estos componentes. A continuación se citan algunos de los servicios de su arquitectura:

- **Activities y Fragments.** Partes del código que realizan una tarea en concreto llevada a cabo por la aplicación. Lleva asociado una interfaz de gráfica para hacer más fácil la interacción con el usuario.
- **Views.** Clase base para construir la interfaz gráfica para cada Activity o Fragment.





- **Notification Manager.** Mecanismo que se encarga de gestionar las notificaciones que se muestran en la barra de notificaciones para interactuar con el usuario.
- **Content Providers.** Reúne un conjunto de datos de la aplicación para que estén disponibles para otras aplicaciones.
- **Resource Manager.** Gestiona todos los elementos que están fuera de la parte del código como imágenes, audios o layouts.
- **Intents.** Nos permite indicar a la aplicación que queremos realizar una acción con otro componente de la misma aplicación u otra.

### 2.2.3. Entorno de desarrollo

Una aplicación Android se desarrolla normalmente en el lenguaje Java con un kit para desarrolladores, el Android SDK (Software Development Kit), pero no es una acción exclusiva ya que se dispone de otras herramientas como un kit de desarrollo en C o C++ o hasta una plataforma para crear aplicaciones de forma visual con una serie de bloques, App Inventor, entre otras.

Google pensó en dar a los desarrolladores herramientas para que la creación de aplicaciones en Android fuera más fácil, por ello diseñó el paquete software Android SDK (Software Development Kit). Este conjunto incluye un simulador de teléfono para pruebas, un depurador de código, herramientas para empaquetar las aplicaciones, librerías, documentación y ejemplos de código. Este paquete está en constante actualización para soportar las nuevas versiones de Android que van saliendo a la luz.

Estas herramientas son utilizadas a través de un entorno de desarrollo (IDE). La plataforma oficial es Android Studio anunciada en 2013 y basada en el software IntelliJ IDEA de JetBrains. Este entorno dejó de ser versión de pruebas a finales de 2014 sustituyendo a su anterior IDE oficial Eclipse; programa que aumentaba sus capacidades para programar en Android a través del ADT (Android Development Tools).

#### 2.2.4. Versiones

Las primeras versiones del sistema operativo Android se iniciaron en Septiembre del año 2008 al comercializar HTC la versión Android 1.0. A partir de entonces, el sistema ha ido evolucionando, corrigiendo fallos y agregando nuevas funcionalidades.

Como curiosidad cabe destacar que cada una de las versiones de Android hace alusión a nombres de postres, siguiendo además un orden alfabético. A continuación se enumeran las distintas versiones, comentando brevemente las novedades introducidas respecto a la versión anterior.

- Android 1.0 – Apple pie  
El sistema operativo incluía navegador web, cámara, Google Maps, Gmail, YouTube y Android Market.
- Android 1.1 – Banana bread  
Posibilidad de ocultar el teclado en la llamada y posibilidad de guardar los adjuntos de los MMS.
- Android 1.5 – Cupcake  
Teclado virtual, Bluetooth, widgets en la pantalla principal y grabación de vídeo.
- Android 1.6 – Donut  
Pequeña actualización con búsquedas mejoradas y renovación de la cámara y Android Market.
- Android 2.0 – Eclair  
Sincronización de contactos con redes sociales, mejoras de animaciones y nuevas funcionalidades de la cámara como zoom digital, modo de escena y enfoque macro entre otros.
- Android 2.1 – Eclair  
Introduce fondos de pantalla animados y la posibilidad de traducir la voz en texto para cualquier campo de texto. Más tarde se actualizaría para añadir la funcionalidad Multitouch en las aplicaciones.
- Android 2.2 – Froyo  
Notable incremento de la velocidad, Tretheing (compartir la conexión 3G a través de Wi-fi) y soporte con Adobe Flash.
- Android 2.3 – Gingerbread  
Soporte con dispositivos con doble núcleo y distintos tamaños de pantallas, nuevo teclado virtual y manejo por voz.
- Android 3.0 – Honeycomb  
Diseñada especialmente para Tablets la cual incluía un SDK propio, durando poco tiempo al debido coste de dos plataformas independientes.

- Android 4.0 – Ice Cream Sandwich  
Nueva interfaz de usuario con el diseño Holo, unificando el sistema para Tablets y Smartphones. Incluye mejoras multitareas y ActionBar dentro de cada aplicación.
- Android 4.1 – Jelly Bean  
Mejora de la interfaz de usuario y de la barra de notificaciones y una actualización de Google Now bastante importante.
- Android 4.2 – Jelly Bean  
Imágenes panorámicas y soporte multiusuario.
- Android 4.3 – Jelly Bean  
Mejoras notables en la seguridad del sistema y nuevo sistema de notificaciones.
- Android 4.4 – KitKat  
Reducción del uso de memoria para un uso más eficiente, mejoras de conectividad con NFC e importantes mejoras en la interfaz.
- Android 5.0 – Lollipop  
Extensión del sistema a nuevas plataformas como Google Wear, Google TV y Google Card, nuevo sistema de modo de ahorro de la batería, nuevos sensores e interfaz de usuario con el diseño Material Design.
- Android 5.1 – Lollipop  
Soporte para múltiples SIMs, protección de dispositivos ante pérdidas y mejoras de estabilidad y rendimiento.
- Android 6 – Marshmallow  
Incluye Android Pay (nuevo sistema de pago con el móvil), mejoras en la duración de la batería y soporte con Fingerprint Authentication (huellas digitales).

### 2.3. Google

Compañía fundada por dos estudiantes de doctorado de Stanford, Serguéi Brin y Larry Page, los cuales crearon en 1996 su primer motor de búsqueda llamado BackRub, que utilizaba enlaces para determinar la importancia de páginas webs. No fue hasta el 4 de Septiembre de 1998 cuando se fundó la sociedad tal y como la conocemos actualmente con el nombre de Google.



Figura 11: Primer logotipo de la compañía Google

El servicio con el que la empresa empezó sus comienzos fue con un motor de búsqueda en la web propiedad de Alphabet Inc. Su característica más destacada es su fácil usabilidad para la búsqueda de texto en las páginas webs. A día de hoy, la empresa no sólo es un motor de búsqueda si no que está especializada en productos y servicios relacionados con Internet y software.

Para que el motor de búsqueda pueda funcionar en periodos de tiempo muy pequeños, Google organiza sus servidores alrededor de todo el planeta a través de un sistema distribuido para poder entregar resultados lo más pronto que sea posible. Un sistema distribuido es un conjunto de ordenadores separados físicamente pero conectados por una red de comunicaciones. Sus sistemas cumplen con todos los aspectos de seguridad de un sistema distribuido como son la confidencialidad, integridad, disponibilidad, tolerancia y recuperación de fallos, y transparencia, para soportar el desarrollo de nuevas aplicaciones y servicios web. Esto hace que el motor de búsqueda de Google sea el sitio web más visitado a nivel mundial.

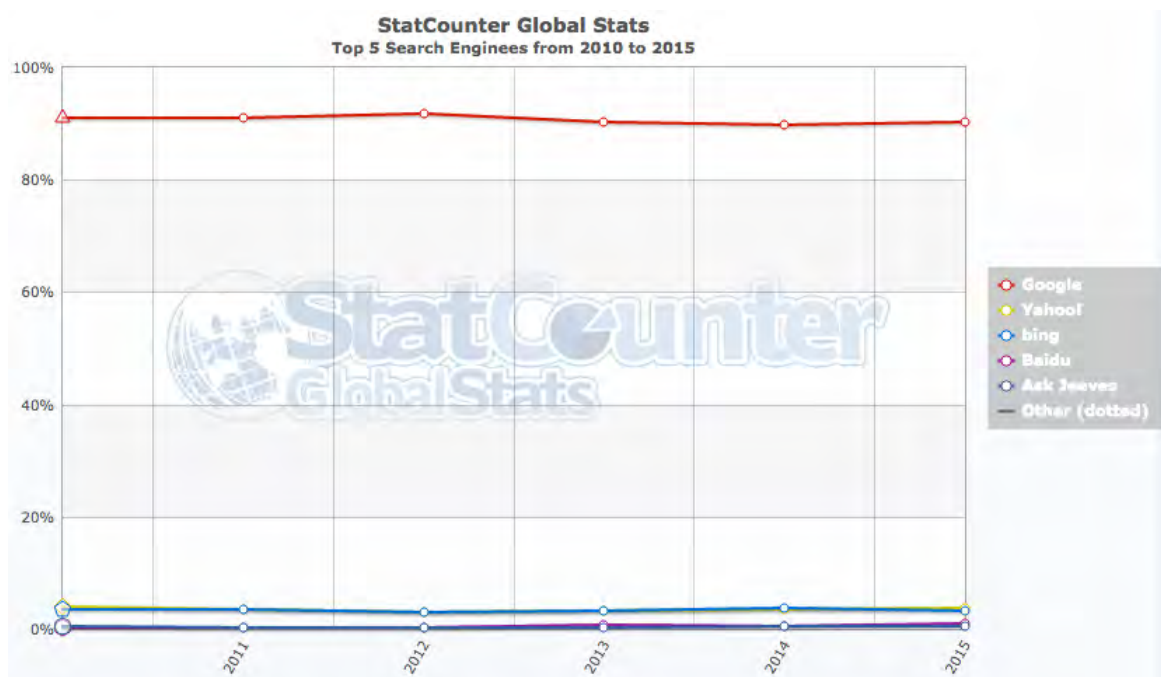


Figura 12: Motores de búsqueda más utilizados a nivel mundial

En la gráfica podemos observar como Google se alza como el buscador más utilizado en los últimos años, dejando muy abajo en la escala a empresas como Yahoo! o Bing; la cual fue creada en 2012 como una alianza entre Microsoft, que disponía de MSN Search, y de Yahoo!, con su motor Yahoo! Search, para intentar hacer frente a Google.

Para tener un hueco en el mercado de software para móviles, en 2005 Google adquirió Android Inc. por 50 millones de dólares, además de contratar a sus fundadores.

En 2006, la empresa Google Inc. adquiere YouTube por 1650 millones de dólares, el portal de descargas de video “online” más utilizado de Internet.

Está considerado el mejor buscador en la historia de Internet, permitiendo un acceso a la información de forma rápida y organizada. Los usuarios de esta herramienta también pueden buscar ciertos contenidos específicos mediante Google Imágenes, Google Video, Google Noticias, etc.

Además de ofrecer el mejor buscador, Google también dispone de otros servicios como:

- Gmail, servicio de correo electrónico gratuito.
- Google Maps, servidor de mapas vía web.
- Google Earth, programa para visualizar imágenes satelitales.
- YouTube, portal de descarga y visualización de videos.
- Google Libros, servicio que busca el texto de los libros que ellos disponen en línea.
- Google Noticias, rastreador de la información de los medios de comunicación “online”.
- Google Chrome, navegador web de código abierto.
- Google Plus, red social a manos de esta compañía.

En 2015 Google cambia su estructura de empresa pasando todas sus operaciones corporativas a una nueva empresa matriz bajo el nombre de Alphabet. Esta nueva sociedad engloba a todas las divisiones que anteriormente tenía en su poder Google, incluyendo los servicios de búsqueda, Maps, Gmail, Chrome, etc.



Figura 13: Logotipo de Google en 2015



### 2.3.1. Google Developers

Es el sitio web donde los programadores pueden encontrar herramientas, interfaces de programación de aplicaciones (APIs) y todos los recursos necesarios para el desarrollo de su software con algún servicio de Google. Creado en el 2005, también dispone de foros de discusión y blogs para que la programación sea más llevadera.

Existen APIs para casi todos los servicios de Google, desde YouTube, Google Maps, Android, etc.

### 2.3.2. Google Maps

Servidor de aplicaciones de mapas vía web proporcionando una potente cartografía fácil de usar. Ofrece desde imágenes de los mapas, fotografías panorámicas por satélite del mundo, rutas entre diferentes ubicaciones, información de localización y contacto de negocios, hasta imágenes a pie de calle con Google Street View. Se promocionan como los mapas más usados en el mundo con más de 205 millones de usuarios, sólo de aplicaciones móviles, activos.

El servicio más usado es el de rutas. Con él se pueden crear rutas para recorrer conduciendo, a pie o en bicicleta. Además dispone de un generador de rutas de transporte y vuelos, los cuales te indican horarios y mejores caminos para llegar a tiempo a su destino. Permite elegir evitar ciertos caminos en la ruta, como peajes o autopistas y elegir a qué hora deseas llegar.

Una herramienta bastante importante es la API que Google desarrolló para Google Maps donde facilita la integración y personalización de los mapas a los desarrolladores de páginas webs o aplicaciones móviles. Con ella, se puede llegar a superponer datos propios sobre un mapa de Google personalizado.

El API se compone de código JavaScript el cual contiene las clases y métodos que se usan para utilizar los mapas en nuestro producto.

### 2.3.3. Google Play

Es la plataforma oficial de aplicaciones y juegos Android desarrollada por Google, además de una tienda digital para la descarga de libros, revistas, películas, música y otros productos de la compañía. Hasta el 2012 llamada Android Market, la herramienta superó a principios de 2015 a su mayor rival App Store (tienda de Apple) en cuanto a número de aplicaciones y número de desarrolladores registrados.



Como desarrollador, el usuario puede subir a este servicio sus aplicaciones o juegos Android de manera gratuita o de pago. Para publicar un producto deberemos de iniciar sesión con una cuenta de Google y realizar un pago de 25 dólares. El dinero que gane el desarrollador por sus aplicaciones será repartido a partes del 70% de beneficio para el usuario y 30% para Google.

## 2.4. Backend as a service

El almacenamiento en la nube es un modelo de almacenamiento de datos basado en el alojamiento de los mismos en espacios virtualizados. Normalmente el proveedor del servicio pone estos datos a disposición del usuario a través de Internet realizando una conexión con su servidor.

Al hablar de proveedores en la nube, podemos citar varios tipos de modalidades que han surgido según la necesidad del usuario.

- El Software as a Service (SaaS) es una aplicación para el usuario final donde paga por un uso de él en la nube. Google Docs, Zoho o Office365 son algunos de ellos.
- El servicio Infraestructure as a Service (IaaS) ofrece un medio de entrega de almacenamiento básico en la red. Las empresas de Amazon y Microsoft proporcionan esta gestión del servicio en la nube.
- El siguiente, Platform as a Service (PaaS) proporciona, además del IaaS, un servidor donde se ejecutarán las aplicaciones del usuario y una base de datos.
- Para facilitar a los desarrolladores el uso del motor en la nube para sus aplicaciones móviles o web, el servicio PaaS evolucionó al servicio Backend as a Service (BaaS).

Los servicios BaaS no sólo facilitan la vinculación de las aplicaciones con el almacenamiento en la nube, sino que también proporcionan características como la gestión de usuarios, notificaciones push, o la integración con las redes sociales, entre otras. Los desarrolladores necesitan unificar de forma rápida y sencilla el front-end (parte del software que interactúa con el usuario) de sus aplicaciones con unos servicios de backend (parte que procesa la entrada desde el front-end) específicos, por ello los proveedores BaaS ofrecen las características citadas anteriormente a través de la utilización de kits personalizados de desarrollo de software (SDK) e interfaces de programación de aplicaciones (API). Así el desarrollador no tiene la necesidad de construir su propio back-end y puede centrarse en el diseño de la interfaz.



Con el crecimiento del número de dispositivos móviles en los últimos años, cada vez más proveedores Baas están surgiendo para satisfacer la demanda de los usuarios. Actualmente existen numerosos proveedores tales como Stackmob, Kinvey, Cloudmine, Parse o BassBox. La mayoría de estas compañías ofrecen un paquete gratuito por el cual se pueden usar algunos de sus servicios en la nube de forma limitada, pero también incluyen otros servicios. Un tipo de modelo por el cual este mercado genera ingresos es el pago de una cuota según el número de usuario o llamadas al servicio al mes, disponiendo además de otro modelo por el que se paga una cuota fija para tener un uso ilimitado.

## 2.5. Vehículo eléctrico

La mayoría de vehículos que están en circulamiento son de combustión interna, siendo los emisores de la mayor fuente de contaminación en las ciudades. Este medio de transporte es vital para las sociedades desarrolladas, apareciendo nuevos conceptos de transportes para la reducción del problema medioambiental y la no contribución al efecto invernadero. Las nuevas alternativas son los vehículos híbridos, que ya están bastante implantados en la sociedad y los vehículos eléctricos, que gracias a las nuevas tecnologías son la mejor alternativa.

El vehículo eléctrico aporta grandes beneficios en materia de eficiencia energética, reducción del uso de productos derivados del petróleo y reducción de emisiones de CO<sub>2</sub>, otras emisiones contaminantes y del efecto invernadero. Por ello este tipo de vehículo ayudan a mejorar la calidad del aire y a la disminución de la contaminación acústica, un problema cada vez mayor en las ciudades.

Según la Real Academia Española, para comprender mejor el significado de vehículo eléctrico podemos definir su significado con su sustantivo “vehículo”,

Vehículo: medio de transporte de personas o cosas

y con el significado de su adjetivo “eléctrico”,

Eléctrico: que funciona mediante electricidad.

Un vehículo eléctrico se puede definir como un medio de transporte que funciona mediante la electricidad. Según esta definición, este tipo de vehículo engloba una amplia cantidad de tipos de medios de transporte, no solo diferenciando la forma ni el medio por el que se desplazan si no también la forma de obtener la energía eléctrica para funcionar:

- Trenes eléctricos, que usan una fuente externa.
- Vehículos solares, donde usan energías renovables.
- Embarcaciones, que generan su energía a bordo.



En este proyecto nos centraremos en turismos que usan la energía eléctrica para propulsarse ya que, cada vez más, son los más cercanos al usuario.

Desde la aparición del primer vehículo autopropulsado a vapor, hace casi dos siglos, hasta nuestros días, nos encontramos que los vehículos eléctricos han ido apareciendo poco a poco en la historia como una sucesión de intentos fallidos y resurgían en alguna crisis de petróleo o diferentes guerras. Los primeros modelos datan de los años 30 del siglo XIX, a manos de Robert Anderson con el primer vehículo eléctrico puro, con baterías no recargables.

En la segunda mitad del siglo XIX aparecieron nuevos modelos de vehículos eléctricos al inventarse la batería recargable. A partir de entonces, la investigación en estos modelos arrancó, y llevó en 1899 a Camille Jenatzy establecer un record mundial en velocidad, superando los 100 km/h con un coche eléctrico.



Figura 14: Vehículo eléctrico del siglo XIX

Con un bajo ruido, su nueva velocidad y una autonomía y coste aceptables para la época, los coches eléctricos empezaron a superar en ventas a los de gasolina. Sin embargo, al inventar Henry Ford en 1912 las cadenas de montaje masivas y baratas, un precio de la gasolina bastante bajo, mayor velocidad en vehículos de combustión y carreteras pensadas para viajes largos, hicieron que la venta de vehículos eléctricos cayera, desapareciendo por completo sobre el 1930.

El motor eléctrico ha poseído siempre unas ventajas, que a día de hoy siguen siendo las mismas:

- Poca contaminación acústica.
- Poca contaminación atmosférica.
- Robustez.
- No utiliza una materia prima limitada, como el petróleo.



- Mantenimiento y coste mucho menor.
- Fácil usabilidad.

A pesar de estas ventajas, presentaba unas limitaciones tecnológicas:

- Tiempo de carga de las baterías demasiado largos.
- Prestaciones limitadas.
- Poca autonomía.
- Algunos tipos de vehículos seguían contaminando al utilizar electricidad proveniente de materias primas contaminantes.

Por estas razones, el motor de combustión tuvo más éxito en la historia.

A partir de los años 60, por la conciencia de la ecología de las personas, empezaron a aparecer varios modelos eléctricos, tales como el tranvía eléctrico, coches de golf eléctricos y pequeños automóviles urbanos. Con la primera crisis del petróleo en 1973, se empezaron a buscar soluciones alternativas, forzando a los fabricantes de coches a mejorar la eficiencia de los utilitarios convencionales por si los países productores del petróleo podían hundir la economía mundial si empezaba a encarecerse. Fueron varios países los que apostaron en tecnología eléctrica y en sus puntos de recarga. A pesar de esto, los modelos no tuvieron mucho éxito ya que usaban baterías de níquel-cadmio (Ni-Cd) con una autonomía limitada a 60/80 kilómetros.

En la época de 1990, apareció un nuevo tipo de motor como alternativa al coche eléctrico, el vehículo híbrido. Los motores híbridos son una combinación de motor térmico con un motor eléctrico, existiendo algunos modelos que no necesitan conectarse a una toma de corriente eléctrica para recargar su batería. Esto es posible al generador en el sistema de frenos que mantienen la carga en la batería. Con las populares ventas del Toyota Prius híbrido en 1998, muchas marcas empezaron a hacerse un hueco en este mercado.

La tecnología de litio no llegó hasta 1996 al mundo del automóvil en un prototipo de Nissan. Esta nueva alternativa proporcionaba al vehículo una mayor estabilidad, no teniendo efecto memoria y aumentando a una autonomía mucho mayor, pudiendo llegar hasta los 300 km.

En la actualidad la tecnología de los vehículos eléctricos está resurgiendo gracias al desarrollo de las nuevas baterías de iones de litio (con las que se supera los 450 km de autonomía) y a la concienciación de los estados en el calentamiento global y recursos petrolíferos, con bastantes ayudas de los gobiernos para que esta vez este tipo de transporte pueda prosperar.

## 2.6. Movele

Movele es el nombre de un Plan de Acción del Ministerio de Industria, Energía y Turismo que se enmarca dentro de la Estrategia Integral de Impulso al Vehículo Eléctrico en España. Para promover el desarrollo y utilización de los vehículos eléctricos en España se facilita la adaptación de la infraestructura eléctrica para la recarga y gestión de los mismos, impulsando programas de información y comunicación de esta tecnología.



Figura 15: Logotipo del Plan Movele

Entre las medidas que está tomando el Ministerio están las ayudas para la adquisición de estos vehículos. Desde principios del año 2013, estos programas de incentivos han contribuido a favorecer la compra de más de 10.000 vehículos eléctricos en España hasta finales de 2014. Esto permitirá a lo largo de su vida útil, evitar la emisión de 150.000 toneladas de CO<sub>2</sub> y el ahorro de 300.000 barriles de petróleo.

Las ayudas dadas para el Programa Movele 2014 se agotaron antes de terminar el año, lo que nos hace ver el cada vez mayor interés de este mercado. Por su gran acogida, este año se ha vuelto a tener una nueva edición del programa, Plan Movele 2015, estableciéndose en los puntos de venta a partir de abril de 2015 hasta finales de año. La financiación de ayudas para este año alcanza un importe de 7.000.000 euros proveniente del Ministerio de Industria, Energía y Turismo.

Al igual que los planes PIVE, se aporta un descuento equivalente a la subvención pública por parte de los fabricantes e importadores que quieran adherirse al programa Movele 2015. Además de este incentivo, salvo renuncia del comprador del vehículo, se instala un punto de carga vinculado al beneficiario.

La ayuda añade a vehículos comerciales, autobuses y cuadríciclos, siendo de una media de 5.500 euros, aproximadamente según el tipo de turismo, para los coches habituales del mercado.

El Gobierno de España, ha puesto a disposición 761 puntos de recarga de uso público, existiendo de carga normal, de carga rápida, de motocicletas y de vehículos adaptados para minusválidos. Para acceder a ellos de forma gratuita se debe de solicitar una tarjeta para activar la recarga del vehículo, según cada comunidad autónoma. La tarjeta Movele da derecho a electricidad para la recarga del vehículo de forma gratuita.



Figura 16: Punto de recarga Movele

Desde finales de 2014 y abril de 2015, periodo que cubre el tiempo entre que se agotaron las ayudas y la aprobación de la nueva edición del programa, los postes estuvieron bastante olvidados no estando todos perfectamente operativos. Para que esto no volviera a ocurrir, han concedido la gestión y mantenimiento de estos a varias empresas, comenzando trabajos de remodelación y actualización de los puntos de recarga actual e instalación de nuevos. Se está a la espera de las nuevas condiciones técnicas y operativas de uso, viendo en un futuro no muy lejano que las recargas dejen de ser totalmente gratuitas, manteniendo un descuento en las mismas.

Desde la página web de Movele (<http://www.movele.es/>) se pueden localizar los puntos de recarga de vehículos eléctricos disponibles en España dentro del programa.



## Capítulo 3. Análisis de los sistemas

Este capítulo hace referencia, en forma de requisitos, a las capacidades, funcionalidades y restricciones de los sistemas.

Se facilita la estructura que va a presentar la aplicación para presentar todos los detalles de la misma. Con ello se intentará cubrir todas las necesidades del usuario respecto al uso de una aplicación del mismo estilo. Además se recogen las diferentes funcionalidades a las que podrá optar un usuario, estableciendo la base sobre la que se desarrolla el diseño, analizando las alternativas surgidas durante el desarrollo del proyecto, argumentando la solución finalmente escogida.

### 3.1. Requisitos

A continuación se enumeran todos los requisitos de la aplicación, capacidades y necesidades que se consideran para el correcto funcionamiento de la misma.

#### 3.1.1. Requisitos Comunes De Los Interfaces

##### a) Interfaces de Usuario

La interface será intuitiva y fácil de usar. El texto y los botones serán claramente definidos, para que cualquier usuario pueda utilizarlo sin problemas. Lo primero que aparecerá al abrir la aplicación será la pantalla de login. Después de iniciar sesión, nos mostrará el mapa con la posición actual del usuario y botones de fácil comprensión.

##### b) Interfaces de Hardware

Dispositivo Android con una versión mínima 4.0.3 (API 15, Ice Cream Sandwich). No necesitará mucha memoria del teléfono ya que las tablas de datos se guardarán en un servicio en la nube.

##### c) Interfaces de Software

El programa se implementará con el sistema Android para dispositivos móviles.



### 3.1.2. Requisitos funcionales

En este epígrafe se enumeran todos los requisitos funcionales que definen el comportamiento del sistema que se va a desarrollar. Además, se valorará el tipo y prioridad de éstos.

Para ofrecer una mejor comprensión, cada requisito se define en una tabla. Cada tabla se formará de los siguientes atributos:

- **Identificador.** Cada requisito debe vincularse a un identificador único para un seguimiento futuro más simple. Con el objeto de tener mayor claridad en su identificación se utiliza la siguiente notación:  
Siglas: Se refiere al tipo de requisito.  
Número: Especifica el número de requisito.

Un ejemplo de requisito funcional es RF-XX; RF significando “requisito funcional” y XX especificando el número de requisito, es decir, para el primer requisito funcional el identificador será “RF-01”.

- **Nombre.** El nombre del requisito que debe ser bastante descriptivo.
- **Tipo.** La importancia de un requisito para que esté implementado en la aplicación o no, tomando los valores de obligatorio u opcional.
- **Prioridad.** La preferencia de un requisito frente a otros. Los requisitos de más prioridad deben implementarse antes que los demás, pudiendo tomar los valores de alta, media y baja prioridad.
- **Descripción.** El detalle de lo que debe hacer el requisito. No debe de ser muy extensa, proporcionando de forma correcta en qué consiste el requisito.

En la siguiente tabla se representa una plantilla que expone un requisito funcional general.

Identificador	RF-XX		
Nombre			
Tipo	<input type="checkbox"/> Obligatorio		<input type="checkbox"/> Opcional
Prioridad	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción			

Tabla 1: Plantilla requisito funcional

Los requisitos funcionales definidos en la aplicación son los siguientes:

Identificador	RF-01		
Nombre	Identificación de usuario		
Tipo	<input type="checkbox"/> Obligatorio	<input checked="" type="checkbox"/> Opcional	
Prioridad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El usuario se podrá identificar a través la pantalla de login.		

Tabla 2: Requisito funcional (I)

Identificador	RF-02		
Nombre	Registro de usuario		
Tipo	<input type="checkbox"/> Obligatorio	<input checked="" type="checkbox"/> Opcional	
Prioridad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El usuario se podrá registrar para poder usar la aplicación.		

Tabla 3: Requisito funcional (II)

Identificador	RF-03		
Nombre	Mapa		
Tipo	<input checked="" type="checkbox"/> Obligatorio	<input type="checkbox"/> Opcional	
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Consulta del mapa desde la posición actual. Es la pantalla principal de la aplicación, donde se accede a todas las opciones.		

Tabla 4: Requisito funcional (III)

Identificador	RF-04		
Nombre	Buscador de destino		
Tipo	<input checked="" type="checkbox"/> Obligatorio	<input type="checkbox"/> Opcional	
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Gestión de la ruta a realizar.		

Tabla 5: Requisito funcional (IV)

Identificador	RF-05		
Nombre	Barra de información		
Tipo	<input checked="" type="checkbox"/> Obligatorio	<input type="checkbox"/> Opcional	
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Muestra las estadísticas de la ruta indicada.		

Tabla 6: Requisito funcional (V)

Identificador	RF-06		
Nombre	Búsqueda de puntos de recarga		
Tipo	<input type="checkbox"/> Obligatorio	<input checked="" type="checkbox"/> Opcional	
Prioridad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Búsqueda de los puntos de recarga del municipio donde se encuentra el usuario.		

Tabla 7: Requisito funcional (VI)

Identificador	RF-07		
Nombre	Garaje		
Tipo	<input checked="" type="checkbox"/> Obligatorio	<input type="checkbox"/> Opcional	
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Almacenamiento de datos de los vehículos que normalmente utiliza el usuario.		

Tabla 8: Requisito funcional (VII)

Identificador	RF-08		
Nombre	Concesionario		
Tipo	<input checked="" type="checkbox"/> Obligatorio	<input type="checkbox"/> Opcional	
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Lista de vehículos eléctricos actualmente en el mercado.		

Tabla 9: Requisito funcional (VIII)

Identificador	RF-09		
Nombre	Menú		
Tipo	<input type="checkbox"/> Obligatorio	<input checked="" type="checkbox"/> Opcional	
Prioridad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Ajustes de la aplicación, información y cierre de sesión de usuario.		

Tabla 10: Requisito funcional (IX)



### 3.1.3. Requisitos no funcionales

A continuación se describirán los requisitos no funcionales que tienen que ver con restricciones y exigencias de calidad del sistema.

Del mismo modo que en el apartado anterior, se establecen tablas en las cuales se indicará el identificador, el nombre, tipo, prioridad del requisito y una breve descripción. Se utiliza la siguiente notación:

- RN-XX; RF significando “requisito no funcional” y XX especificando el número de requisito, es decir, para el primer requisito funcional el identificador será “RN-01”.

Los requisitos no funcionales definidos en la aplicación son los siguientes:

Identificador	RN-01		
Nombre	Diseño		
Tipo	<input checked="" type="checkbox"/> Obligatorio	<input type="checkbox"/> Opcional	
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Se usará un diseño similar a todas las aplicaciones actuales para que sea intuitiva y fácil de usar.		

Tabla 11: Requisito no funcional (I)

Identificador	RN-02		
Nombre	Programación		
Tipo	<input checked="" type="checkbox"/> Obligatorio	<input type="checkbox"/> Opcional	
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	La aplicación se programará en Java bajo el sistema Android.		

Tabla 12: Requisito no funcional (II)

Identificador	RN-03		
Nombre	Hardware		
Tipo	<input checked="" type="checkbox"/> Obligatorio	<input type="checkbox"/> Opcional	
Prioridad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Dispositivo Android con una versión mínima 4.0.3 (API 15, Ice Cream Sandwich).		

Tabla 13: Requisito no funcional (III)

Identificador	RN-04		
Nombre	Base de datos		
Tipo	<input checked="" type="checkbox"/> Obligatorio	<input type="checkbox"/> Opcional	
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Se utilizará una base de datos en un servicio en la nube para un mejor control de la aplicación.		

Tabla 14: Requisito no funcional (IV)

Identificador	RN-05		
Nombre	Equipo de trabajo		
Tipo	<input checked="" type="checkbox"/> Obligatorio	<input type="checkbox"/> Opcional	
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Se utilizará al menos un ordenador para desarrollar el proyecto y elaborar la documentación.		

Tabla 15: Requisito no funcional (V)

Identificador	RN-06		
Nombre	Entorno de desarrollo		
Tipo	<input checked="" type="checkbox"/> Obligatorio	<input type="checkbox"/> Opcional	
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Como IDE (Entorno de Desarrollo Integrado) se utilizará Android Studio para la plataforma Android.		

Tabla 16: Requisito no funcional (VI)

Identificador	RN-07		
Nombre	Espacio de trabajo		
Tipo	<input checked="" type="checkbox"/> Obligatorio	<input type="checkbox"/> Opcional	
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Se dispondrá de un espacio lo suficientemente amplio para albergar el equipo de trabajo.		

Tabla 17: Requisito no funcional (VII)

Identificador	RN-08		
Nombre	Ubicación		
Tipo	<input checked="" type="checkbox"/> Obligatorio	<input type="checkbox"/> Opcional	
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Determinación de la ubicación del usuario a través GPS, Wi-Fi o redes móviles.		

Tabla 18: Requisito no funcional (VIII)

Identificador	RN-09		
Nombre	Librerías adicionales		
Tipo	<input checked="" type="checkbox"/> Obligatorio	<input type="checkbox"/> Opcional	
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Como librerías adicionales se necesitarán las que proporciona Google para la parte de mapas y geolocalización, Parse para el servicio en la nube y Jsoup para leer html.		

Tabla 19: Requisito no funcional (IX)

Identificador	RN-10		
Nombre	Servicio de puntos de recarga		
Tipo	<input type="checkbox"/> Obligatorio	<input checked="" type="checkbox"/> Opcional	
Prioridad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Se utilizará un servicio público para la localización de los puntos de recarga de vehículos eléctricos en España.		

Tabla 20: Requisito no funcional (X)

Identificador	RN-11		
Nombre	Servicio de mapas		
Tipo	<input checked="" type="checkbox"/> Obligatorio	<input type="checkbox"/> Opcional	
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Se usará los servicios de Google Maps para la representación de los mapas.		

Tabla 21: Requisito no funcional (XI)

## 3.2. Casos de uso

En este apartado se dispone de un esquema representativo de los casos de uso. Este esquema representa las funciones que el usuario puede llevar a cabo con la aplicación que se ha desarrollado.

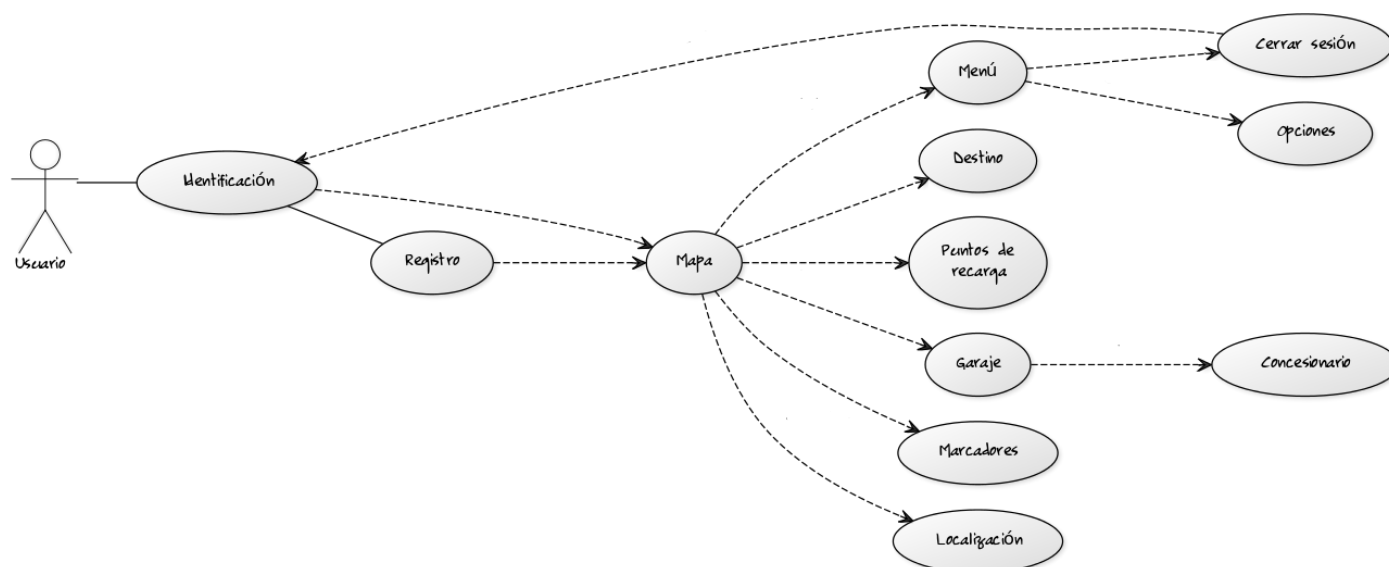


Figura 17: Representación esquemática de los casos de uso

A continuación se explica cada uno de los casos de uso representados en el esquema anterior. Al igual que se hizo para los requisitos, se hará uso de tablas para especificar cada caso, las cuales se compondrán de los siguientes campos:

- Caso de uso. Nombre que resume la función del caso de uso a definir.
- Objetivo. Describe la acción que realiza el caso de uso.
- Escenario previo. Acción la cual hace que el caso de uso actúe.
- Escenario final. Acción que realiza el caso de uso.

CASO DE USO	IDENTIFICACIÓN
Objetivo	Iniciar sesión en la aplicación para obtener los datos de usuario.
Escenario previo	Apertura de la aplicación.
Escenario final	Mapa.

Tabla 22: Caso de uso (I)

CASO DE USO	REGISTRO
Objetivo	Registro del usuario para poder acceder a la aplicación
Escenario previo	Identificación
Escenario final	Mapa

Tabla 23: Caso de uso (II)

CASO DE USO	DESTINO
Objetivo	Introducción de un nuevo destino para realizar una búsqueda
Escenario previo	Mapa
Escenario final	Mapa

Tabla 24: Caso de uso (III)

CASO DE USO	PUNTOS DE RECARGA
Objetivo	Muestra los puntos de recarga disponibles en el municipio de nuestra ubicación
Escenario previo	Mapa
Escenario final	Mapa

Tabla 25: Caso de uso (IV)

CASO DE USO	GARAJE
Objetivo	Vista de los vehículos guardados por el usuario
Escenario previo	Mapa
Escenario final	Garaje

Tabla 26: Caso de uso (V)

CASO DE USO	CONCESIONARIO
Objetivo	Lista de los vehículos eléctricos existentes
Escenario previo	Garaje
Escenario final	Concesionario

Tabla 27: Caso de uso (VI)

CASO DE USO	MARCADORES
Objetivo	Puntos de selección encima del mapa
Escenario previo	Mapa
Escenario final	Google Maps

Tabla 28: Caso de uso (VII)

CASO DE USO	MENÚ
Objetivo	Visión de los ajustes de la aplicación
Escenario previo	Mapa
Escenario final	Menú

Tabla 29: Caso de uso (VIII)

CASO DE USO	OPCIONES
Objetivo	Ajustes de la aplicación
Escenario previo	Menú
Escenario final	Mapa

Tabla 30: Caso de uso (IX)

CASO DE USO	ACERCA DE...
Objetivo	Información sobre la aplicación
Escenario previo	Menú
Escenario final	Mapa

Tabla 31: Caso de uso (X)

CASO DE USO	CERRAR SESIÓN
Objetivo	Cerrar la sesión del usuario actual para poder identificarse con otro usuario distinto.
Escenario previo	Menú
Escenario final	Identificación

Tabla 32: Caso de uso (XI)



CASO DE USO	LOCALIZACIÓN
Objetivo	Centrar la localización actual del usuario en el mapa
Escenario previo	Mapa
Escenario final	Mapa

Tabla 33: Caso de uso (XII)

### 3.3. Análisis de alternativas de la solución

Tras el estudio de las necesidades, funcionalidades y restricciones del proyecto, se analizan las posibles alternativas que han surgido al realizar la aplicación.

#### 3.3.1. Alternativas del servicio de puntos de recarga

En un principio, al pensar en vehículos eléctricos se asocia a vehículos corrientes que recargan sus baterías como repostan carburante los otros vehículos. Por ello se pensó en las gasolineras del ámbito español. El Ministerio de Industria, Energía y Turismo dispone de un portal en el cual podemos consultar todas las estaciones de servicio existentes. Este servicio se llama Geoportal, disponible en <http://geoportalgasolineras.es/>, y nos devuelve todas las estaciones de servicio de España con la siguiente información:

- Provincia
- Localidad
- Dirección
- Rótulo
- Latitud y longitud
- Horario de apertura
- Tipo de carburante que dispone
- Precio del carburante actualizado cada hora

Este servicio era idóneo para nuestra aplicación, pero tiene un problema, de momento no está incluida la información de los puntos de recarga en estaciones de servicio. Esto es debido a que este portal está pensado en un primer momento para consultar el precio del carburante de cada estación. Además, los puntos de recarga suelen instalarse en sitios diversos como parking, aeropuertos, en plena calle, en centros comercial y también en estaciones de servicio, entre otros muchos lugares.

Se empezaron a buscar alternativas pero sólo se encontraron aplicaciones y páginas webs privadas que muestran los puntos de recarga desde una base de datos propia donde, además del propietario, los usuarios pueden añadir localizaciones donde se encuentran dichos puntos y cuyos servicios no son accesibles por terceros.





El tutor de este proyecto tenía en conocimiento la existencia de una página web del Ministerio donde en otras ocasiones él había consultado información sobre puntos de recarga en España, [www.movele.es](http://www.movele.es). Al acceder a este portal subvencionado por las Administraciones Públicas, se observó que, aunque sólo muestra los puntos de recarga públicos y gratuitos dentro del programa Movele, cada año proponen una nueva edición de subvenciones las cuales hacen crecer y mejorar este plan.

Esta última alternativa del Plan Movele, se descartó en un primer momento, ya que no se dispone de un servicio como para poder ser usado por aplicaciones como la propuesta en este proyecto. Por ello se pensó en utilizar el servicio de Geoportal, utilizando la consulta de un tipo de carburante cualquiera en plan simulación para el proyecto y en un futuro, esperando una actualización con los puntos de recarga, sólo tener que cambiar el tipo de consulta.

Después de realizar la aplicación con el servicio de Geoportal, se pensó en una manera de leer la información del portal Movele teniendo así una aplicación de la cual el usuario puede tener un uso real desde un primer momento y no sólo valer para la realización del proyecto.

La elección de la solución fue parsear la página web de Movele, realizando un filtro por municipio antes de su lectura. Con esta solución obtenemos todas las gasolineras de España que están dentro del Programa Movele, actualizado al Plan Movele 2015, siendo la mejor opción para llevar a cabo el proyecto.

### 3.3.2. Alternativas de la base de datos

Para poder consultar los datos de los vehículos a usar por el usuario era necesario elegir un lugar donde almacenarlos. Se plantearon dos opciones posibles.

La primera alternativa ponía en uso una base de datos interna, la cual almacenaría y gestionaría la información dentro de la propia aplicación. Por otro lado se tenía una segunda alternativa, la cual era un servicio en la nube para guardar todos los datos necesarios.

La opción de una base de datos interna en la aplicación supondría que para cada vez que se quisiera añadir un nuevo vehículo del mercado a esa base de datos se tendría que actualizar la aplicación, además de ampliar el tamaño de la aplicación cuanto más datos tuviera. Como actual usuario de un dispositivo Smartphone, el tener que actualizar muchas veces una aplicación para mínimos cambios llega a ser muy pesado, lo que hace que una aplicación pueda ser desinstalada para no volver a usarla buscando otras aplicaciones alternativas.

Como para el uso de las funciones principales de la aplicación, los cuales son buscar una ruta o los puntos de recarga, se necesita de internet, la segunda opción de guardar los datos en un servicio de la nube fue la elección de la solución seleccionada.

## Capítulo 4. Diseño de la solución técnica

En el presente capítulo se exponen las funcionalidades más importantes de la aplicación analizando de forma detallada su lógica.

### 4.1. Base de datos

Para el desarrollo de la aplicación se utiliza una base de datos desde un servicio en la nube. Se decidió el servicio Baas (Backend as a Service) que ofrece la empresa Parse (<https://www.parse.com/>) por ser gratuita, con los servicios que utiliza la aplicación, y por su sencillez en uso frente a otras compañías gracias a su SDK.

En la aplicación disponemos de los usuarios que se han dado de alta, tabla Users y la lista de vehículos eléctricos disponibles en el mercado, tabla Cars. Al pensar en la relación entre los datos podemos ver que un usuario puede tener varios vehículos en su poder y un modelo de vehículo lo pueden haber comprado varios usuarios. Por lo tanto estamos ante una relación  $n$  a  $n$  entre ambos conceptos en la cual aparecerá una tabla intermedia para gestionar la relación, a la cual se denomina Garage.

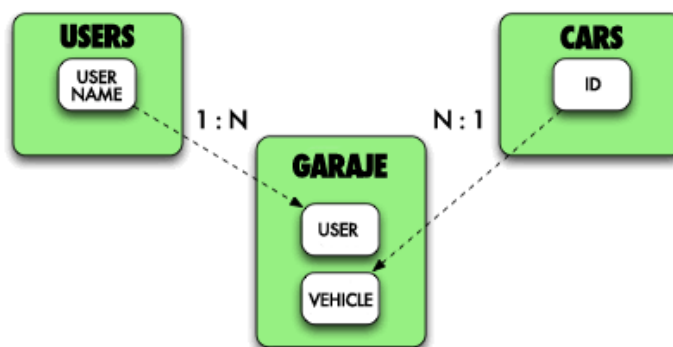
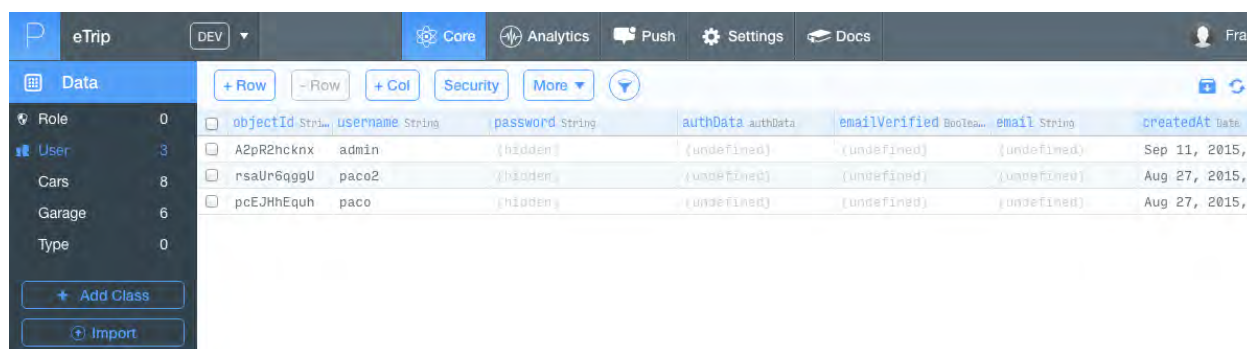


Figura 18: Relación entre tablas

Estamos ante un diagrama entidad relación con tres tablas denominadas Users, Cars y Garage.

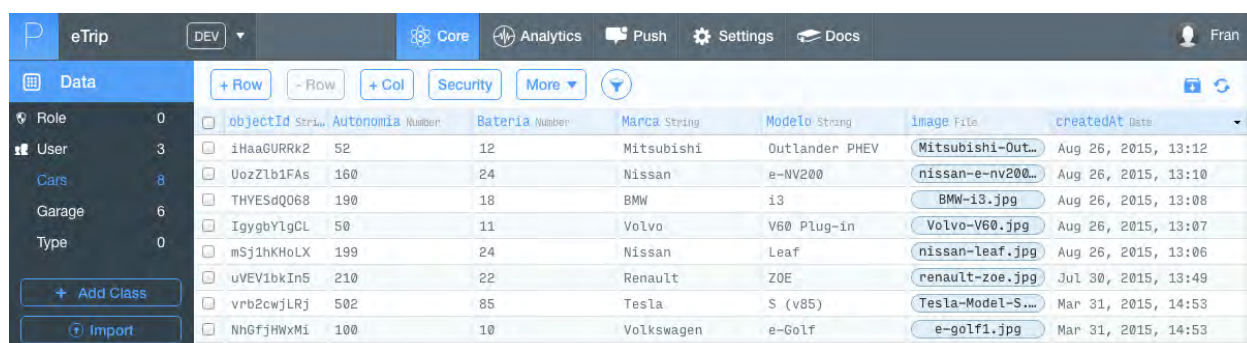
La primera de las tablas (Users) se encuentran los usuarios dados de alta en la aplicación. En ella se guarda el nombre introducido por el usuario y su contraseña que por seguridad es un campo que no se muestra al administrador del servicio de Parse. Esta tabla se decidió que contuviera tan pocos datos para tener un registro de la aplicación sencillo y que no asuste al usuario con multitud de campos ni verificaciones de correo electrónico como hacen otras aplicaciones.



objectId	username	password	authData	emailVerified	email	createdAt
A2pR2hcknx	admin	[hidden]	{undefined}	{undefined}	{undefined}	Sep 11, 2015,
rsaUr6gggU	paco2	[hidden]	{undefined}	{undefined}	{undefined}	Aug 27, 2015,
pcEJHhEqh	paco	[hidden]	{undefined}	{undefined}	{undefined}	Aug 27, 2015,

Figura 19: Tabla "Users" de Parse

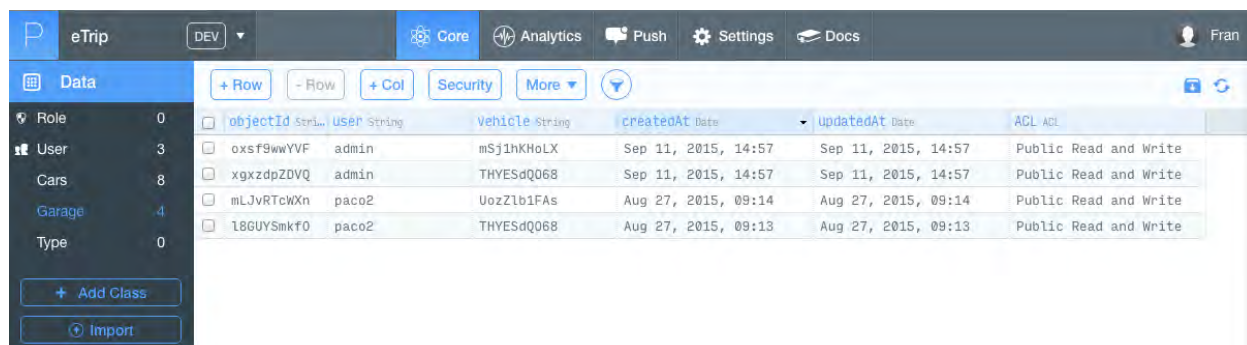
En la segunda de las tablas (Cars) se tiene la lista de vehículos eléctricos que el usuario puede elegir para su uso. De cada uno se dispone la marca y modelo del vehículo, su autonomía en kilómetros y la capacidad de la batería en kilómetros hora mostrada en su ficha técnica oficial, y una imagen del vehículo. Estos datos son suficientes para mostrar la información necesaria en la aplicación.



objectId	Autonomia	Bateria	Marca	Modelo	image	createdAt
iHaaGURRk2	52	12	Mitsubishi	Outlander PHEV	Mitsubishi-Out...	Aug 26, 2015, 13:12
UozZlb1FAs	160	24	Nissan	e-NV200	nissan-e-nv200...	Aug 26, 2015, 13:10
THVESdQ068	190	18	BMW	i3	BMW-i3.jpg	Aug 26, 2015, 13:08
IgygbYlgCL	50	11	Volvo	V60 Plug-in	Volvo-V60.jpg	Aug 26, 2015, 13:07
mSj1hKHolX	199	24	Nissan	Leaf	nissan-leaf.jpg	Aug 26, 2015, 13:06
uVEV1bkIn5	210	22	Renault	ZOE	renault-zoe.jpg	Jul 30, 2015, 13:49
vrh2cwjLRj	502	85	Tesla	S (v85)	Tesla-Model-S...	Mar 31, 2015, 14:53
NhGfjHWxM1	100	10	Volkswagen	e-Golf	e-golf1.jpg	Mar 31, 2015, 14:53

Figura 20: Tabla "Cars" de Parse

La última de las tablas (Garage) almacena los vehículos que dispone cada usuario. Cada vez que se elige un vehículo de la lista del Concesionario, guardamos en esta tabla su nombre de usuario y el identificador del vehículo para luego mostrarlos en la lista del Garage.



objectId	user	vehicle	createdAt	updatedAt	ACL
oxsf9wwYVF	admin	mSj1hKHolX	Sep 11, 2015, 14:57	Sep 11, 2015, 14:57	Public Read and Write
gxzdpZDVQ	admin	THVESdQ068	Sep 11, 2015, 14:57	Sep 11, 2015, 14:57	Public Read and Write
mLJvRTcWxN	paco2	UozZlb1FAs	Aug 27, 2015, 09:14	Aug 27, 2015, 09:14	Public Read and Write
18GUYSmkF0	paco2	THVESdQ068	Aug 27, 2015, 09:13	Aug 27, 2015, 09:13	Public Read and Write

Figura 21: Tabla "Garage" de Parse



Los datos se guardan en forma clave-valor de cadenas alfanuméricas, siendo texto o número en nuestro caso. El proceso de guardar y leer datos de las tablas de Parse se realiza en segundo plano (background).

No se ha utilizado la parte de almacenamiento en el teléfono ya que las tablas disponen de pocos datos, siendo rápida su descarga. Los elementos que más pueden ocupar son las imágenes, pero se realiza un almacenamiento de las mismas en la memoria caché del teléfono para tener un acceso más rápido a las imágenes. Si la imagen ha sido descargada anteriormente, la aplicación usará la imagen almacenada anteriormente en la memoria para ser mostrada.

## 4.2. Google Maps API

En este capítulo se expone una guía orientativa con los pasos necesarios para incluir y utilizar el servicio de Google Maps en el desarrollo de una aplicación móvil.

### 4.2.1. Google Play Services

Para poder incluir en nuestro proyecto la API (Application Programming Interface) de Google Maps debemos instalarnos los Google Play Services que nos proporcionan las herramientas necesarias para poder trabajar con el servicio de mapas, además de otros servicios. Para ello, abrimos el SDK Manager y nos aseguramos que tenemos descargados estos servicios de Google. También podemos descargar Android Support Repository y Google Repository, librerías y recursos de soporte que podríamos usar en nuestra aplicación para incluir funcionalidades nuevas dando soporte a APIs antiguas. Todos estos servicios los podremos encontrar dentro de la carpeta Extras en el SDK Manager.

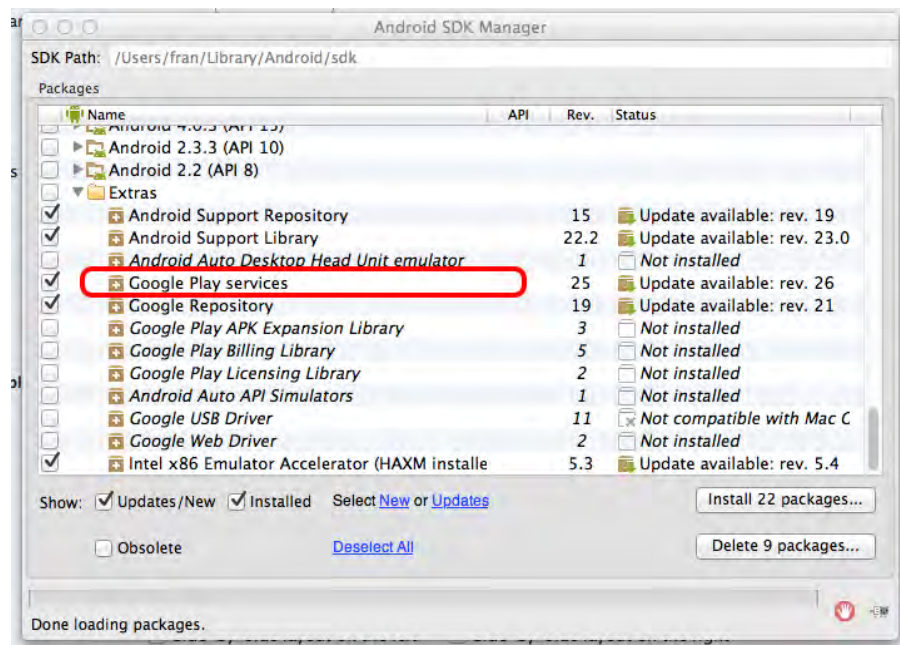


Figura 22: SDK Manager de Android

Cuando los tengamos descargados e instalados debemos importar esta librería a nuestro proyecto añadiendo la siguiente línea al archivo build.gradle:

```
dependencies {
    compile 'com.google.android.gms:play-services:7.0.0'
}
```

El número final depende de la versión que esté disponible en ese momento.

#### 4.2.2. API Key de Google

Una vez instalado Google Play Services necesitamos una clave para acceder a los servicios de Google, la API Key. Esta clave se utiliza para autenticar las solicitudes al servicio a efectos contables. Ya que algunos servicios de Google, como son los mapas, tienen un número de peticiones a partir del cual deja de ser un servicio gratuito.

Los usuarios estándar de la API disponen de:

- 2.500 solicitudes de rutas al día.
- 8 puntos de recorrido permitidos en cada solicitud.
- 10 peticiones por segundo.

A partir de estas solicitudes, se tiene la posibilidad de pagar 0.50\$ por cada 1.000 solicitudes adicionales, hasta 100.000 al día. Si se quieren mejores servicios se debe contratar una cuenta Premium cuya cuota varía según lo contratado. Nuestra aplicación no supera el límite gratuito por lo que una API Key estándar nos permite usar el servicio sin ninguna limitación.

Para conseguir nuestra API Key debemos de acceder a la Consola de desarrolladores de Google, <https://console.developers.google.com/project>. Debemos crear un nuevo proyecto, asignándole un nombre y, si lo preferimos, modificar el identificador del proyecto.

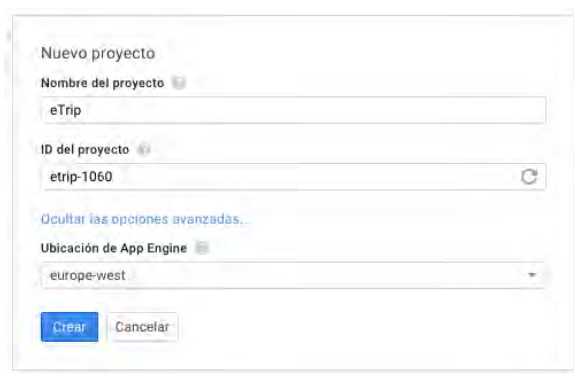


Figura 23: Creación de un proyecto nuevo

Una vez creado, nos aparecerá la consola de nuestro proyecto donde debemos de navegar hasta APIs y autenticación/APIs, para habilitar las siguientes APIs:

- Google Maps Android API v2.
- Google Maps Coordinate API.
- Google Maps Engine API.
- Google Maps Geolocation API.
- Translate API.



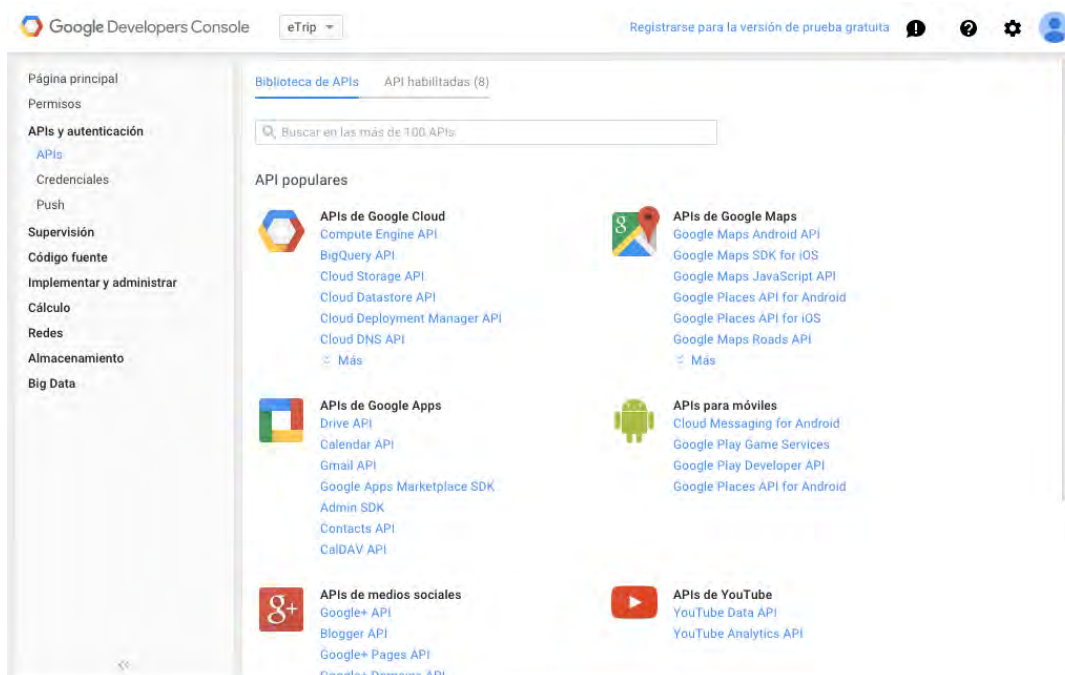
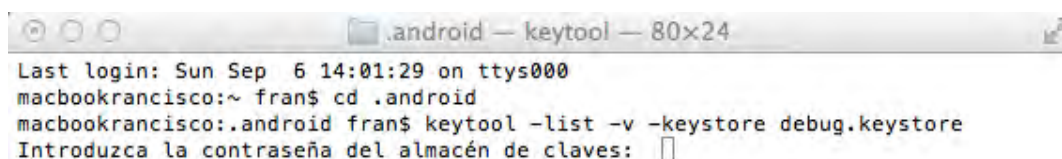


Figura 24: Consola de desarrolladores de Google

Una vez activadas, debemos crear las credenciales necesarias para utilizar estas APIs, en el apartado APIs y autenticación/Credenciales de la consola. Como mínimo se deben crear dos credenciales, una para el uso de la aplicación como desarrollador y otra para la versión en producción que se generará en un archivo apk. La credencial como desarrollador estará vinculada al ordenador donde ejecutemos la aplicación en modo de pruebas, por lo que se necesitarán tantas credenciales como puestos de trabajo estemos usando.

Añadimos una credencial para una clave de API, indicando que sea una clave de Android en nuestro caso. Para ello nos pide introducir un nombre, para poder identificarla entre las demás claves, el nombre del paquete principal de la aplicación y una huella digital de certificado SHA1 con la que firmaremos la aplicación. El nombre del paquete lo encontramos en el archivo AndroidManifest.xml del proyecto en Android Studio. La huella para obtener la clave para desarrollador se encuentra en un fichero llamado debug.keystore en la carpeta oculta llamada “.android” de nuestra carpeta personal. Abrimos un terminal, nos dirigimos a esa carpeta oculta e introducimos el comando

```
keytool -list -v -keystore debug.keystore
```



```
.android — keytool — 80x24
Last login: Sun Sep 6 14:01:29 on ttys000
macbookrancisco:~ fran$ cd .android
macbookrancisco:.android fran$ keytool -list -v -keystore debug.keystore
Introduzca la contraseña del almacén de claves: 
```

Figura 25: Comandos clave debug en un terminal iOS

donde la contraseña por defecto es “android” sin comillas. Copiamos el valor de la huella SHA1 y la introducimos en la consola para desarrolladores de Google. Al realizar estos pasos, Google nos proporcionará nuestra clave de API (API Key), la cual debemos pegar en su lugar correspondiente en el AndroidManifest.xml de nuestro proyecto de Android Studio.



```
<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="YOUR_KEY_HERE" />
```

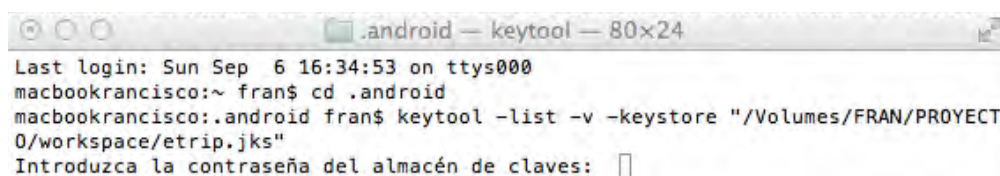
Figura 26: Parte de código del archivo AndroidManifest.xml

El último paso para poder visualizar el mapa en nuestra aplicación es indicar en el Manifest que la API v2 de Google Maps utiliza OpenGL ES versión 2, introduciendo el siguiente código:

```
<uses-feature
    android:glEsVersion="0x00020000"
    android:required="true"/>
```

Los pasos anteriores son para cuando queremos depurar, si ahora queremos publicar la aplicación tendremos que crearnos una nueva API Key para ello. Cuando generamos un apk, utilizamos un Key Store (Almacen de claves) propio, el cual es distinto al anteriormente nombrado “debug.keystore”. Repetiremos los pasos anteriores pero esta vez nombrando el nuevo keystore en la línea de código del terminal e introduciendo la contraseña elegida al crear este Key Store.





```
.android — keytool — 80x24
Last login: Sun Sep  6 16:34:53 on ttys000
macbookrancisco:~ fran$ cd .android
macbookrancisco:.android fran$ keytool -list -v -keystore "/Volumes/Fran/PROYECTO/workspace/etrip.jks"
Introduzca la contraseña del almacén de claves: 
```

Figura 27: Comandos clave producción en un terminal iOS

Una buena práctica, es comentar el archivo Android Manifest con los API Keys y SHA1 utilizados normalmente, ya que deberemos cambiar entre ellos cada vez que utilicemos un ordenador de trabajo distinto o generemos un apk del proyecto.

### 4.2.3. Servicio de rutas

El servicio de rutas de Google Maps es facilitado a los desarrolladores realizando una petición al servicio con la siguiente dirección:

<https://maps.googleapis.com/maps/api/directions/>

El servicio nos puede devolver los datos en formato json y xml, recomendándose json al ser un formato ligero y simple de leer para el intercambio de datos.

Se requieren ciertos parámetros para la obtención de los resultados de la ruta, mientras que otros son opcionales, separándose cada parámetro del signo &. A continuación se citan los parámetros y valores utilizados en la aplicación, al igual que un ejemplo para su mayor comprensión:

<https://maps.googleapis.com/maps/api/directions/json?origin=40.332362,%20-3.765531&destination=valencia&mode=driving&avoid=tolls&region=es&language=es>

- origin. Valor de origen de la ruta, ya siendo coordenadas, latitud y longitud o formato textual con el nombre del lugar.
- destination. Valor de destino de la ruta, al igual que el anterior puede ser en coordenadas o texto.
- sensor. Para indicar si la aplicación utiliza un sensor del teléfono para determinar la ubicación del usuario. Este valor ya no es necesario incluirlo.

- mode. Indica el modo de transporte a usar en la petición de las direcciones, pudiendo ser driving, walking, bicycling o transit. Para nuestra aplicación es necesario que este parámetro sea del tipo conducción (driving).
- avoid. Especifica las rutas que debe evitar al realizar el cálculo, eligiéndose en el menú de opciones de la aplicación. Este parámetro puede indicar tres alternativas, evitar peajes, autopistas y ferris, ya siendo individualmente o las tres juntas separándose por el símbolo |. Por ejemplo: *avoid=tolls/highways/ferries*.
- region. Valor del código de región donde queremos hacer la búsqueda. Al utilizar esta aplicación en España este parámetro es muy útil, ya que existen nombres de ciudades que no sólo se encuentran en nuestro país, como por ejemplo Guadalajara. Indicando el código "es", le estamos proporcionando información de a qué país nos queremos referir.
- language. Especifica el idioma en el que se nos proporcionarán los resultados, siendo en nuestro caso "es".

A continuación se detalla la respuesta de la petición al indicarle los parámetros del ejemplo anterior, explicando los parámetros que han sido utilizados para la aplicación.

```
{
  "geocoded_waypoints" : [
    {
      "geocoder_status" : "OK",
      "place_id" : "ChIJP2WtCluJQQ0RooYHxKQ8TBI",
      "types" : [ "route" ]
    },
    {
      "geocoder_status" : "OK",
      "place_id" : "ChIJb7Dv8ExPYA0ROR1_HwFRo7Q",
      "types" : [ "locality", "political" ]
    }
  ],
  "routes" : [
    {
      "bounds" : {
        "northeast" : {
          "lat" : 40.3716028,
          "lng" : -0.3757857
        },
        "southwest" : {
          "lat" : 39.429081,
          "lng" : -3.7678183
        }
      },
      "copyrights" : "Datos de mapas ©2015 Google, Inst. Geogr. Nacional",
      "legs" : [
        {
          "distance" : {
            "text" : "363 km",
            "value" : 363479
          },
          "duration" : {
            "text" : "3h 24 min",
            "value" : 12242
          },
          "end_address" : "Valencia, Valencia, España",

```



```

    },
    "via_waypoint" : []
  }
},
"overview_polyline" : {
  "points"
  :
  "mduF~q~U@zHkUcOqWcHwO_n@os@qW_Fwl@jg@}IA|j@g\\|YxDhTaSx\\wyAHgvCa|@c_Ckv@otAqi@o}AiLkgA}}kA_hA{hA{Eog@zl@ebC
  nx@sbC`u@{z@z~@uq@hkAgpBr^uU|Rkx@bW_`A|oA{oAlQyoAlHk[~a@kZlkAoi@f_@_q@pi@i}D~AwR@oT}q@jGik@~s@wsAu@}z@iMwj@I
  d@yi@dg@u^xg@yM|p@r@`YoW~Rgm@dk@ua@do@g_@fUsr@hr@aNtn@mrA~gBy^hj@wr@nYmjA`NiBAl_@iz@lq@ekDIAuq@tk@geBvN
  mrBhb@{u@po@ejAdSg{ @tx@u|@p{AoqAl`BsjCzaA_uB~jB}{Bnj@qf@dZel@zxA{sBxnAgb@ho@kQbz@sx@`d@w}C{FquAcDu_@|Qag@h}@a
  k@zm@q~@|kAalCvcA_`Djol@f|qy@nyAawEnv@mdEvs@_iBcAugEbHqhCxVqdChkByvDb`@geB@o{ @_Ban@zUgo@b\\ezAl[quB|NwjDfz@or
  DbTelDfFmnAneAkr@|jAgdAzUciBbbCcoD`c@mf@`r@o~Bpf@ol@vPsLdMeAfaAb\\`XzL||AyWn~Akr@tk@o`An~@k|z|@wz@`jAwy@n{`e@j
  u@q`hb@_}@txAuc@tx@u}AlXurArc@ex@tcAmyAdnA__DpyAor@`rBwia`aA}D`cAfQjs@{Uzl@udAb`A{UxbB_aBrXaxAdZugAb`A{s@tzAkfCh`@
  wmBwXavA_ZacAuDgxAfc@wrAlfAmwAha@q~AcGqlAmh@uxDo|w}AgQwoBk}@uiBchAamA{k@oq@kR_}AyCobCop@afByi@wwAgGuxAhXo|B
  wEokCjDy{A|g@c|CIDupCb`@iyBnPksg~XcrB`QagCKolEq`@oeBgLmtA~MipAsN{uAc`A{sA}MklCzu`Bsa@awAcnAcBmQeuA~g@cv@f~@onDfb
  @ilA~AmvA_D}nApM}jAmp@sdEaKojDqk@gqC~JukAeL_hAgj@gfAoH}oA{w@kwDkMozEt@aqAtn@ihAdmB{pAni@obAxlgua|g@{wAx|@s~@b
  c@epA`a@odChqAgtC`t@qyBjDobAhjBu`MjNelBj^yy@bPmq@cHq^gJ_yAyCibAm}juzBflpAd|ic@nf@k~B|Xu`And@cgAdg@ek@llocBwY{|Bc\\g
  kAuo@sqBjy@srDk`AkoAg_@mpA}LotIVq}NfAoeHyKktCaj@ivDlkAsaFzGkaD~G{uAwQk`@iBX"
  },
  "summary" : "A-3",
  "warnings" : [],
  "waypoint_order" : []
}
},
"status" : "OK"
}

```

- status. Para saber si la solicitud ha sido correcta.
- routes. Con este elemento accedemos a legs y step, ya que están anidados en su interior.
- distance. Para indicar al usuario la distancia total de la ruta.
- duration. Contiene el valor de la duración total de la ruta.
- legs. Contiene los datos con los tramos de la ruta. Si sólo se indica un destino, como en este caso, sólo dispondremos de un leg.
- steps. Nos indican una serie de pasos con la información del tramo de la ruta.
- polyline. Valor codificado con una serie de puntos que representan los pasos que hay que dar para la correcta representación de la ruta.

### 4.3. Parsear página web

La página web del Plan Movele no dispone de un servicio como tal para poder ser utilizado para nuestra aplicación, por ello se decidió parsear la web en búsqueda de los datos necesarios para la obtención de los puntos de recarga.

La URL utilizada es

<http://www.movele.es/index.php/mod.puntos/mem.mapa/reلمenu.20/filtro.>



a la cual añadimos al final el filtro deseado. En nuestro caso se filtra la información por municipio, así la web nos muestra todas los puntos de recarga disponibles en el municipio indicado.

Para la extracción y manipulación de los datos de la web se utiliza la librería Jsoup, utilizando DOM como método de parseo propio de Android. Con la filosofía DOM, el documento XML se lee completamente antes de interaccionar con él. Es un método más sencillo de usar que otros, como SAX, ya que devuelve su contenido en forma de una estructura en árbol, donde los elementos se dividen en nodos con una jerarquía padre-hijo.

Del resultado del parseo, nos guardamos el nombre identificativo del punto de recarga y un enlace en formato Google Satellite Link, el cual nos envía a otra página. Esta segunda web se vuelve a parsear para recoger la latitud y longitud del punto que nos interesa.

Con estos datos ya podemos añadir a nuestro mapa los marcadores con los puntos de recarga.

#### 4.4. Control de versiones

El control de versiones es un sistema que almacena los cambios realizados a lo largo del tiempo. Con ello se tiene la posibilidad de recuperar una versión específica en cualquier momento.

Es una buena herramienta cuando se trabaja en proyectos de este estilo con varios desarrolladores, ya que cada uno puede trabajar por separado y subir en cualquier momento su versión de código. Si dos personas han estado modificando partes iguales, el sistema avisará para que se revise cuáles son las diferencias entre los dos proyectos.

En el presente trabajo, esto no ha sido necesario, al estar desarrollando un único alumno. Se ha necesitado un control de versiones para hacer copias de seguridad del proyecto según se iba avanzando.

Se ha utilizado la herramienta GitHub como control de versiones ya que te ofrece una cuenta gratuita en sus servidores. La dirección web del repositorio es

<https://github.com/franvara/eTrip>

donde se han ido subiendo las versiones de la aplicación, administrándolas en una rama develop donde se iba almacenando cada cambio y en una rama master donde figura el proyecto terminado.

Para el uso del servicio se ha necesitado un estudio de esta herramienta, ya que anteriormente no se había utilizado. Se ha llegado a la conclusión de que es una herramienta bastante útil para el desarrollo de aplicaciones móviles por todo lo anteriormente comentado. Además se ha comprobado que con el uso de la consola de línea de comandos del sistema se puede tener mayor control sobre los repositorios.

#### 4.5. Formulación

Al indicar un destino y un vehículo eléctrico, la aplicación nos devuelve información sobre esta ruta. Un dato de los incluidos en esta información es el gasto de batería desde el inicio hasta el final de la ruta. Este cálculo se obtiene con la siguiente formulación:

- Primero calculamos el consumo, en kWh/km, del vehículo seleccionado con la capacidad de su batería y la autonomía dados por el fabricante usando un ciclo de conducción NEDC (New European Driving Cycle).

$$\text{Consumo (kWh/km)} = \frac{\text{Batería (kWh)} * 1 \text{ km}}{\text{Autonomía(km)}}$$

- Con el consumo específico de electricidad que gasta cada kilómetro y la distancia de la ruta en km proporcionada por Google Maps, obtenemos el consumo de la ruta elegida, en unidades de kWh.

$$\text{ConsumoRuta(kWh)} = \frac{\text{Consumo} * \text{Distancia}}{1 \text{ Km}}$$

- Suponiendo que gastamos el 100% de la batería, ahora debemos calcular cuánto gasto de la batería tendríamos con el consumo de la ruta calculado anteriormente.

$$\text{GastoBatería(\%)} = \frac{\text{ConsumoRuta} * 100}{\text{Batería}}$$

Para una mejor comprensión, a continuación se presenta un ejemplo:

Ruta: Aranjuez - Valencia

Distancia: 334 km

Vehículo: Renault Zoe, Batería: 22 kWh, Autonomía: 210 km



$$\text{Consumo} = \frac{\text{Batería (kWh)} * 1 \text{ km}}{\text{Autonomía(km)}} = \frac{22\text{kWh} * 1\text{km}}{210\text{km}} = 0'104761\text{kWh/km}$$

$$\text{ConsumoRuta} = \frac{\text{Consumo} * \text{Distancia}}{1 \text{ Km}} = \frac{0'104761\text{kwh/km} * 334\text{km}}{1\text{km}} = 34'99\text{kWh}$$

$$\text{GastoBatería} = \frac{\text{ConsumoRuta} * 100}{\text{Batería}} = \frac{34'99\text{kWh} * 100}{22\text{Kwh}} = 159\%$$

Para el ejemplo anterior necesitaríamos un 159% de batería, lo que equivale a 1'59 cargas de batería completas para realizar esa ruta con ese vehículo en concreto.

Los cálculos no han sido más exactos ya que para ello necesitaríamos más datos de la ruta como son la velocidad máxima o la pendiente de cada tramo, entre otros. Estos datos se pueden obtener en la nueva API Google Maps Roads pero se necesita un plan de pago en Google, el cual no se dispone.

#### 4.6. Versión mínima

Se ha elegido una versión mínima de Android de 4.0.3 (API 15, Ice Cream Sandwich). Dispositivos que tengan una versión menor del sistema no podrán instalarse la aplicación.

Cuanto menor sea la versión mínima, más dispositivos podremos abarcar. El API 15 es la mejor opción ya que el 94 % de los dispositivos que están activos en el Google Play Store usan esta versión u otra superior, además de incluir muchas de las funciones que se usan normalmente en las aplicaciones actuales.



## 4.7. Interfaz de usuario

Uno de los puntos más importantes en la aplicación es el diseño visual ya que debe ser simple para dar una comprensión sencilla del funcionamiento a simple vista. La interfaz gráfica es una parte bastante olvidada por los programadores, teniendo en muchas empresas un departamento de diseño gráfico para realizar este trabajo.

### 4.7.1. Estudio de diseño

Antes de empezar, es necesario realizar un estudio de los componentes de la aplicación para comprender el funcionamiento que dará el usuario de ellos.

La interfaz debe ser sencilla e intuitiva, no ofreciendo dificultad en su uso. Debe ser lo suficientemente simple como para que cualquier usuario pueda utilizar las principales funciones sin ningún problema. Además debe permitir su comprensión rápidamente sin necesidad de razonamiento.

Para que se cumpla esa sencillez, se diseñará la pantalla con el menor número de opciones y botones. Así el usuario no tendrá que buscar a través de toda la aplicación lo que desea ejecutar.

Para conseguir un diseño intuitivo, pueden seguirse los consejos de diseño oficiales de Google, siendo los siguientes algunos de ellos:

- Utilizar una paleta de colores que disponga de colores primarios. Google nos proporciona muestras de color para descargar y hacer pruebas en nuestra aplicación.
- Simbolizar lo que se quiere decir mejor con iconos que con palabras, ya que el usuario se fijará antes en un icono intuitivo que un texto. Por ello deben ser simples y modernos, asegurando legibilidad y claridad, incluso en tamaños pequeños. Se dispone de una gran variedad de iconos de Google para su uso.
- Usar el nuevo diseño, Material Design, basado en objetos materiales, dando un diseño simple y a la vez moderno.
- Tener en cuenta las distintas versiones de Android, diseñando pantallas compatibles para todas las versiones a la que se da soporte, siendo de igual aspecto.



#### 4.7.2. Pantalla de inicio y registro

Lo primero que el usuario se encuentra nada más acceder a la aplicación es la pantalla de inicio de sesión. En ella podemos encontrar un inicio sencillo con los colores principales de la aplicación y una opción para registrarse si no se tiene aún una cuenta de usuario.

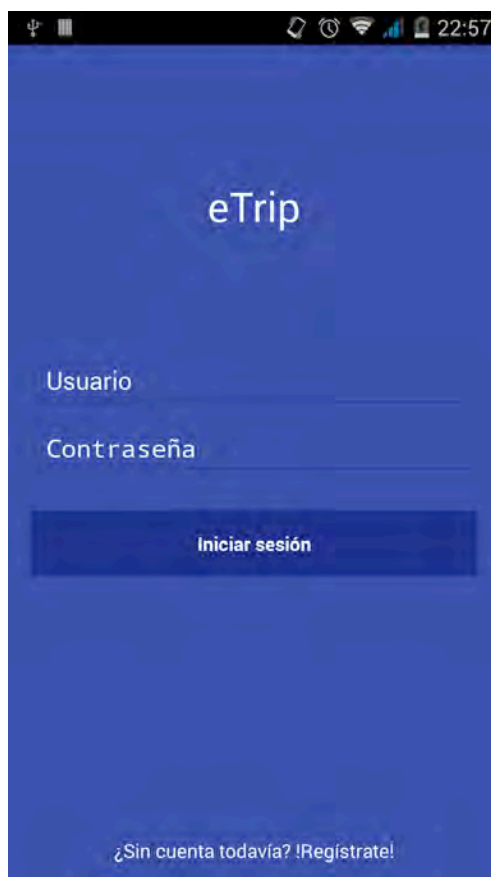


Figura 28: Pantalla de inicio de sesión

Si no se introducen correctamente los datos o si no se tiene conexión a Internet, la aplicación enseña un mensaje indicándoselo al usuario.

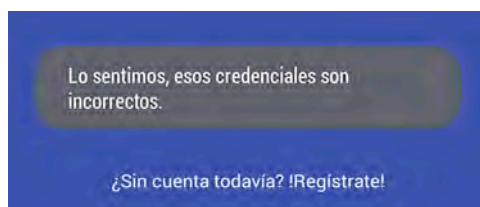


Figura 29: Mensaje de error

En la pantalla de registro se ha optado el disponer de un formulario bastante sencillo, ya que de momento la aplicación no utiliza datos personales de la aplicación, un nombre de usuario y la contraseña siendo repetida en el mismo para que no haya confusiones, es suficiente como datos para pedir al usuario.



Figura 30: Pantalla de registro de usuario

Se han tenido en cuenta todos los mínimos detalles para el diseño de la interfaz. En estas pantallas podemos observar que cuando vamos a introducir los datos, aparece el teclado y toda la pantalla se redimensiona, así no se pierde la visibilidad de ninguna parte de la misma. Por otra parte en el propio teclado podemos navegar hasta la opción para introducir el siguiente dato y en el último campo poder iniciar la acción del botón principal.

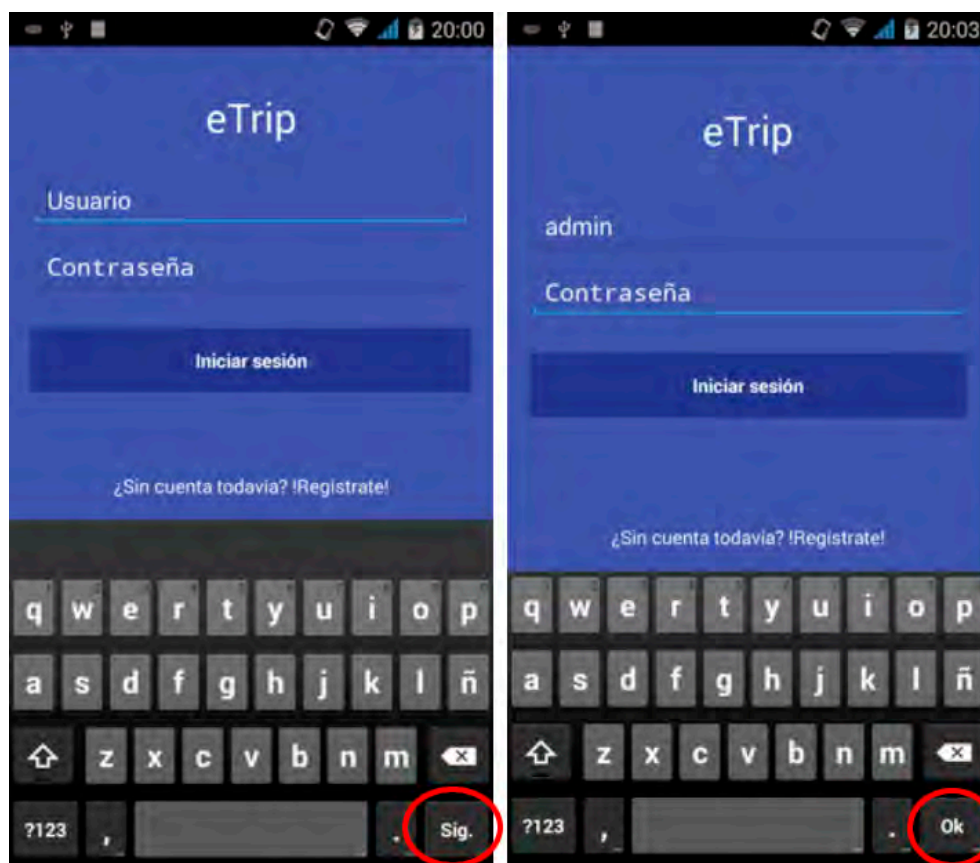


Figura 31: Pantalla de inicio de sesión con teclado

#### 4.7.3. Pantalla principal

En la pantalla principal de la aplicación se representa el mapa. En ella se puede observar una barra de búsqueda sencilla, dos botones flotantes con un diseño moderno e iconos para su fácil comprensión, un cuadro de texto con la información de la ruta elegida y la barra superior (ActionBar) con opciones de menú.

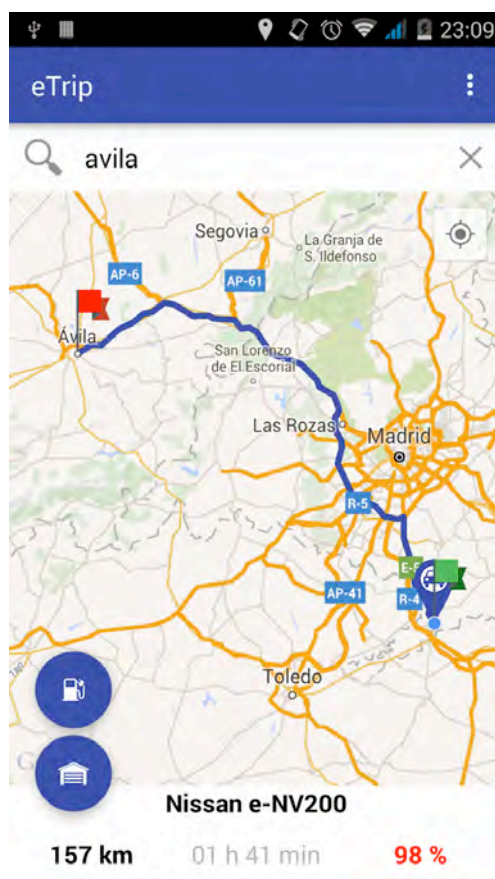


Figura 32: Pantalla principal

En la ActionBar se han tenido en cuenta las distintas versiones del sistema operativo, añadiendo a versiones anteriores a la Android 5.0 Lollipop, una sombra que dispone el nuevo sistema operativo.

Los botones flotantes se pueden ocultar pulsando en cualquier punto del mapa, así tenemos una vista limpia para poder navegar por toda la pantalla.

La mayoría de aplicaciones actuales que están en el mercado, incluyen un menú lateral para disponer de todas las opciones de la aplicación imitando a las de Google en su diseño. En nuestro proyecto se ha eliminado este menú eligiendo uno sencillo ya que disponemos de pocas opciones. Este tipo de menú lateral con pocas opciones al final limita el uso de la aplicación entorpeciendo la visibilidad y eficiencia de la misma, por ello se optó por la sencillez del menú clásico.

#### 4.7.4. Pantallas del garaje y del concesionario

Los vehículos se seleccionan en dos pantallas las cuales están formadas de una lista. Se muestra una fotografía, la marca, modelo y la autonomía de cada vehículo. En el garaje además de poder seleccionar un vehículo y quedarse marcado para ver cuál es el seleccionado, se pueden eliminar cada elemento por separado.

En la Action Bar se ha incluido un botón para ir a la pantalla anterior, realizando la misma opción que hace la acción propia del botón del teléfono y en la pantalla del garaje se ha dispone de un botón para añadir vehículos al garaje personal.

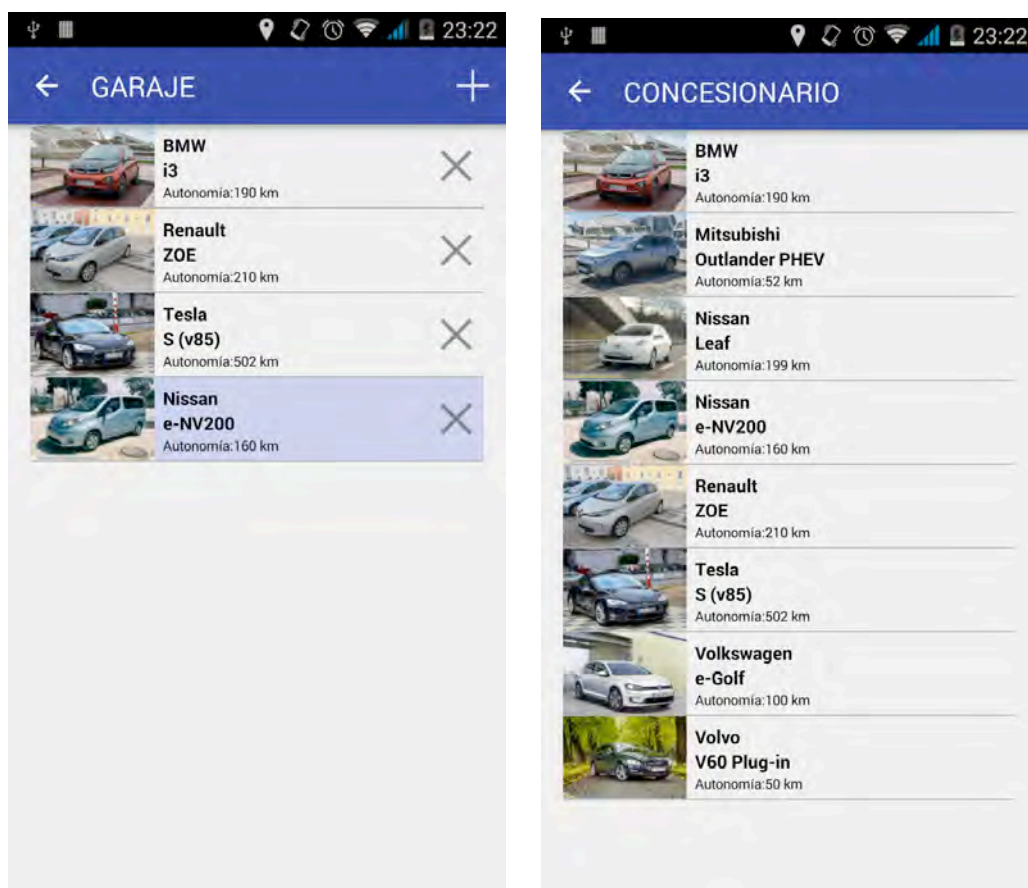


Figura 33: Pantallas de Garaje y Concesionario

#### 4.7.5. Iconos

Para el proyecto se han diseñado todos los iconos de la aplicación para que fueran de fácil comprensión sus acciones.

El icono principal de la aplicación se ha diseñado con un color azul, simulando el azul del logotipo de la universidad, elegido entre la variedad de azules de la paleta de colores recomendada para el diseño del Material Design. El diseño simula una ruta con los marcadores de inicio y fin de la misma.



Figura 34: Icono principal de la aplicación

Al añadir un icono en una aplicación Android hay que tener en cuenta que puede ser instalada en muchos dispositivos, los cuales tendrán diversas resoluciones y tamaños de pantalla. Por ello para que se vea correctamente debemos de tener distintos tamaños de cada icono para que se ejecute el adecuado según el teléfono.

Para obtener los diferentes tamaños existe una herramienta online bastante utilizada por los programadores, <https://romannurik.github.io/AndroidAssetStudio/icons-launcher.html>. Este servicio fue utilizado en la asignatura de la carrera pero se descubrió que no era del todo efectivo ya que realizaba los iconos un poco más pequeños de las plantillas que recomiendan para las aplicaciones de Android.

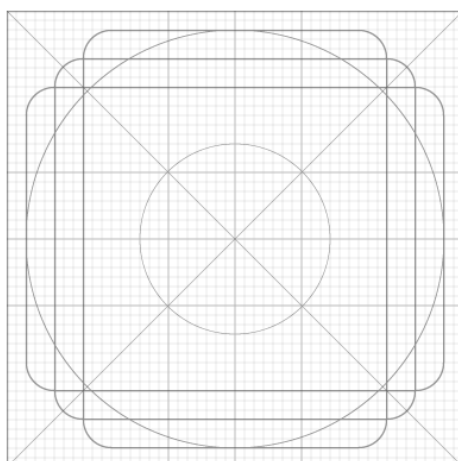


Figura 35: Cuadrícula para iconos de Google

Por estos motivos se diseñó una plantilla propia en Photoshop para realizar todos los tamaños de los iconos para este proyecto. En ella se diseña el icono en máxima calidad y se procede a exportar los diferentes tamaños. También se creó un archivo con extensión .atn para guardar las acciones que se realizan al exportar cada icono, así con solo dar a un botón se exportará cada tamaño para cada icono de la aplicación que hayamos diseñado.



Figura 36: Plantilla iconos

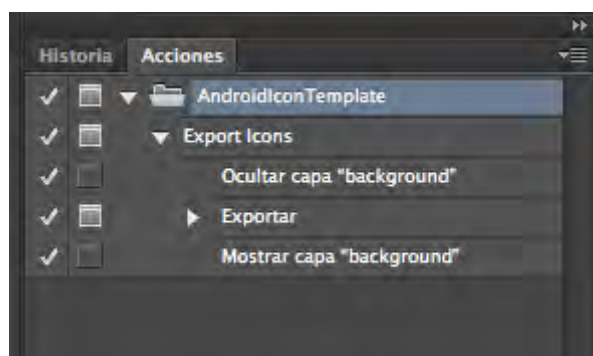


Figura 37: Vista de acciones creadas

De esta forma también se han creado todos los demás iconos de la aplicación, como son los marcadores dentro del mapa.



Figura 38: Iconos del mapa

También se ha tenido en cuenta el caso de si la imagen de cada vehículo no se carga, por algún error en la descarga o si tarda mucho en descargarse según la conexión, se añade una imagen en su lugar hasta que pueda mostrarse la adecuada.



Figura 39: Icono temporal de vehículo



## Capítulo 5. Pruebas de los sistemas y manual de usuario

### 5.1. Pruebas de los sistemas

A continuación se detallan las pruebas realizadas que confirman el correcto funcionamiento de la aplicación. Para ofrecer una mejor comprensión, cada prueba se define en una tabla, formada con los siguientes atributos:

- Identificador. Cada prueba debe vincularse a un identificador único para un seguimiento más simple. Con el objeto de tener mayor claridad en su identificación se utiliza la siguiente notación:

Siglas: Se refiere a la prueba.

Número: Especifica el número de prueba.

Un ejemplo de prueba es P-XX; P significando “Prueba” y XX especificando el número de prueba, es decir, para la primera prueba el identificador será “P-01”.

- Descripción. El detalle de la prueba realizada.
- Test superado. Para realizar la prueba se han utilizado 3 dispositivos reales, con distintas versiones de sistema operativo Android y distintos tamaños de pantalla. Los dispositivos utilizados son: BQ Aquaris E5 HD con Android 4.4.2 y 5 pulgadas de pantalla, Motorola XT1068 con Android 5.0.2 y 5 pulgadas, y Samsung Galaxy Ace 2 (i8160) con Android 4.4.2 y 3’8 pulgadas de pantalla. En esta parte de la tabla se indicará si los dispositivos han superado la prueba.

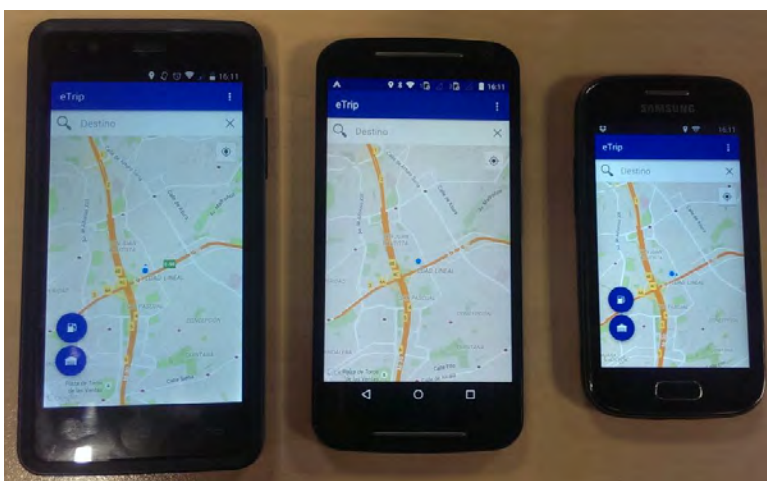


Figura 40: Dispositivos utilizados

En la siguiente tabla se representa una plantilla que expone una prueba general.

Identificador	P-XX		
Descripción			
Test superado	<input type="checkbox"/> BQ	<input type="checkbox"/> Motorola	<input type="checkbox"/> Samsung

Tabla 34: Plantilla de prueba realizada

Las pruebas realizadas para ver el correcto funcionamiento de la aplicación son las siguientes:

Identificador	P-01		
Descripción	Correcta instalación en las diferentes versiones permitidas hasta la versión mínima indicada.		
Test superado	<input checked="" type="checkbox"/> BQ	<input checked="" type="checkbox"/> Motorola	<input checked="" type="checkbox"/> Samsung

Tabla 35: Prueba realizada (I)

Identificador	P-02		
Descripción	Uso de la aplicación sin Internet.		
Test superado	<input checked="" type="checkbox"/> BQ	<input checked="" type="checkbox"/> Motorola	<input checked="" type="checkbox"/> Samsung

Tabla 36: Prueba realizada (II)

Identificador	P-03		
Descripción	Uso de la aplicación sin servicio de ubicación.		
Test superado	<input checked="" type="checkbox"/> BQ	<input checked="" type="checkbox"/> Motorola	<input checked="" type="checkbox"/> Samsung

Tabla 37: Prueba realizada (III)

Identificador	P-04		
Descripción	Inicio de sesión en la aplicación con otros datos.		
Test superado	<input checked="" type="checkbox"/> BQ	<input checked="" type="checkbox"/> Motorola	<input checked="" type="checkbox"/> Samsung

Tabla 38: Prueba realizada (IV)

Identificador	P-05		
Descripción	Registro de usuario en la aplicación con datos ya existentes.		
Test superado	<input checked="" type="checkbox"/> BQ	<input checked="" type="checkbox"/> Motorola	<input checked="" type="checkbox"/> Samsung

Tabla 39: Prueba realizada (V)

Identificador	P-06		
Descripción	Interfaces de usuario visualizadas correctamente en diferentes tamaños de dispositivo.		
Test superado	<input checked="" type="checkbox"/> BQ	<input checked="" type="checkbox"/> Motorola	<input checked="" type="checkbox"/> Samsung

Tabla 40: Prueba realizada (VI)

Identificador	P-07		
Descripción	Introducción de un destino no disponible.		
Test superado	<input checked="" type="checkbox"/> BQ	<input checked="" type="checkbox"/> Motorola	<input checked="" type="checkbox"/> Samsung

Tabla 41: Prueba realizada (VII)

Identificador	P-08		
Descripción	Servicio de Google Maps no disponible.		
Test superado	<input checked="" type="checkbox"/> BQ	<input checked="" type="checkbox"/> Motorola	<input checked="" type="checkbox"/> Samsung

Tabla 42: Prueba realizada (VIII)

Identificador	P-09		
Descripción	Servicio Movele no disponible.		
Test superado	<input checked="" type="checkbox"/> BQ	<input checked="" type="checkbox"/> Motorola	<input checked="" type="checkbox"/> Samsung

Tabla 43: Prueba realizada (IX)

Identificador	P-10		
Descripción	No existen puntos de recargas en el municipio indicado.		
Test superado	<input checked="" type="checkbox"/> BQ	<input checked="" type="checkbox"/> Motorola	<input checked="" type="checkbox"/> Samsung

Tabla 44: Prueba realizada (X)

Identificador	P-11		
Descripción	El servicio de Parse no está disponible para leer la base de datos.		
Test superado	<input checked="" type="checkbox"/> BQ	<input checked="" type="checkbox"/> Motorola	<input checked="" type="checkbox"/> Samsung

Tabla 45: Prueba realizada (XI)

Identificador	P-12		
Descripción	Comprobación de textos correctamente en español e inglés.		
Test superado	<input checked="" type="checkbox"/> BQ	<input checked="" type="checkbox"/> Motorola	<input checked="" type="checkbox"/> Samsung

Tabla 46: Prueba realizada (XII)

Identificador	P-13		
Descripción	Datos de usuario guardados en la memoria del teléfono.		
Test superado	<input checked="" type="checkbox"/> BQ	<input checked="" type="checkbox"/> Motorola	<input checked="" type="checkbox"/> Samsung

Tabla 47: Prueba realizada (XIII)

Identificador	P-14		
Descripción	Datos de vehículo seleccionado guardado en las preferencias del teléfono.		
Test superado	<input checked="" type="checkbox"/> BQ	<input checked="" type="checkbox"/> Motorola	<input checked="" type="checkbox"/> Samsung

Tabla 48: Prueba realizada (XIV)

Identificador	P-15		
Descripción	Ruta pintada correctamente.		
Test superado	<input checked="" type="checkbox"/> BQ	<input checked="" type="checkbox"/> Motorola	<input checked="" type="checkbox"/> Samsung

Tabla 49: Prueba realizada (XV)

Identificador	P-16		
Descripción	Información de ruta obtenida correctamente.		
Test superado	<input checked="" type="checkbox"/> BQ	<input checked="" type="checkbox"/> Motorola	<input checked="" type="checkbox"/> Samsung

Tabla 50: Prueba realizada (XVI)

Para entregar al profesor las versiones del código según se iban implementando y fuera viendo los resultados, se utiliza un sistema de distribución de aplicaciones móviles, Applivery. Esta herramienta es gratuita y está pensada para la distribución de aplicaciones testing tanto para Android como para iOS. La plataforma permite la creación de una página web desde donde se puede acceder a la última versión subida por el desarrollador.

Esta aplicación se puede descargar en [www.applivery.com/etrip](http://www.applivery.com/etrip) o escaneando el siguiente código QR



Figura 41: Código QR para descargar la aplicación

## 5.2. Manual de usuario

Este epígrafe está destinado a una descripción detallada de la aplicación eTrip, enumerando las posibles acciones que el usuario puede realizar.

La aplicación eTrip se presenta bajo el sistema operativo móvil Android destinada a proporcionar un sistema de navegación y gestión de rutas para usuarios que dispongan de vehículos eléctricos. También contempla información de los puntos de recarga proporcionados por el Ministerio a través del plan Movele 2015.

Está basado en un formato de interfaz donde navegar a través de un mapa, en el cual el usuario, habiendo seleccionado su tipo de vehículo eléctrico, puede conocer datos sobre la ruta elegida.

La primera vez que accedemos a la aplicación móvil debemos de iniciar sesión en nuestra cuenta, si no disponemos aún de cuenta en eTrip, debemos de registrarnos en un sencillo cuestionario sólo introduciendo el nombre de usuario y la contraseña que elijamos.

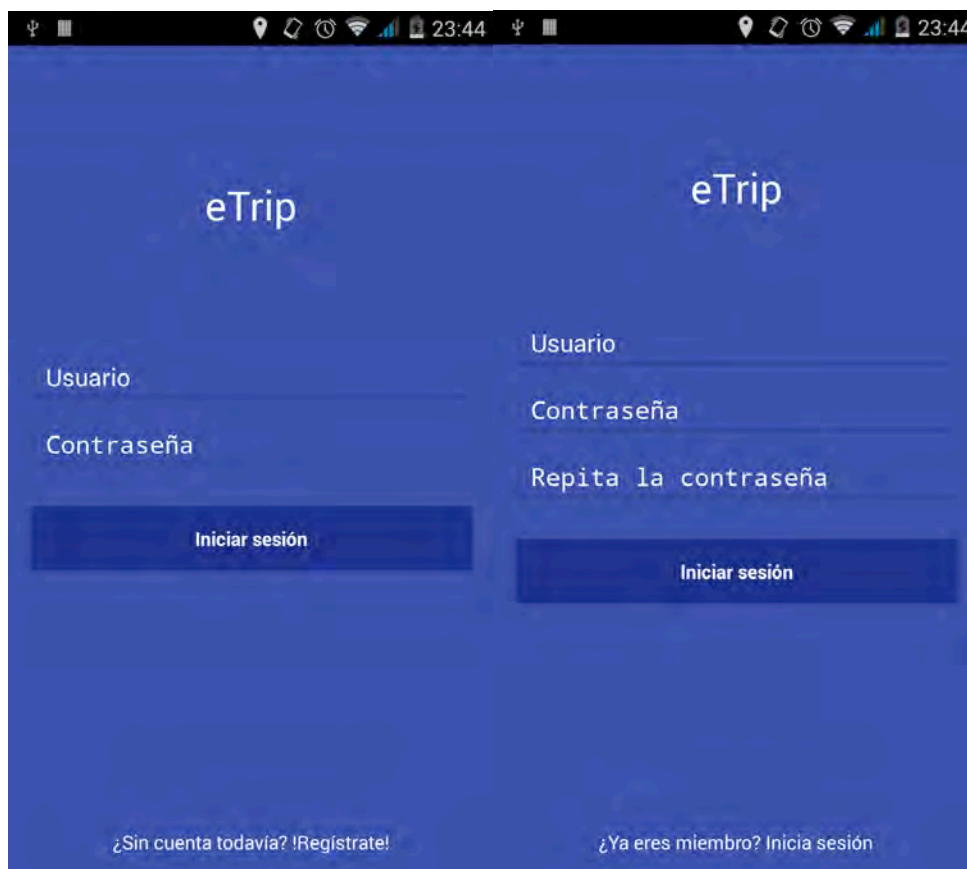


Figura 42: Pantallas de inicio de sesión y registro de usuario

Iniciamos sesión para poder acceder a la aplicación y aparecerá la pantalla principal. En ella podemos utilizar los servicios de la aplicación.

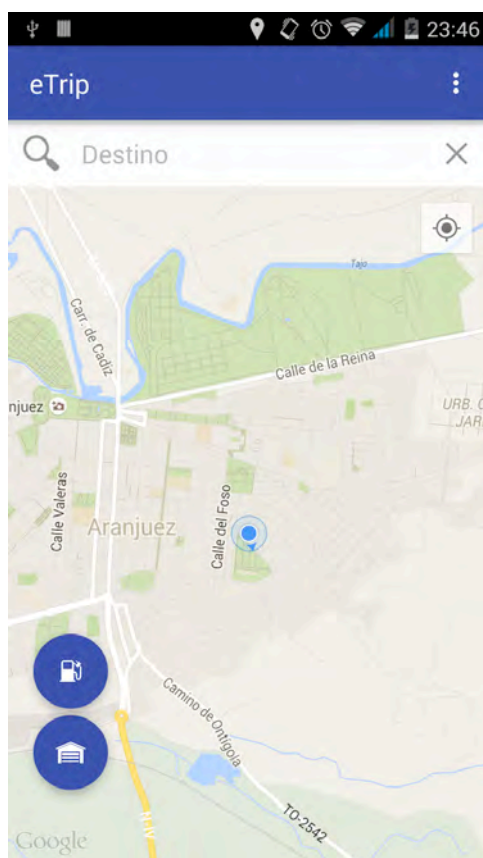


Figura 43: Pantalla principal

Lo primero que podemos ejecutar es la elección de nuestro vehículo. Para ello seleccionaremos el botón con el símbolo de un garaje situado en la parte inferior izquierda.



Figura 44: Botón Garaje



Estando en nuestro garaje podemos añadir tantos vehículos como queramos de la base de datos accediendo desde el botón “+” de la parte superior derecha de la pantalla.



Figura 45: Sección Garaje

En la sección “Concesionario” tenemos una lista con los vehículos eléctricos disponibles en el mercado.





Figura 46: Sección Concesionario

Al seleccionar uno de ellos se añadirá a nuestro garaje. Podemos tener en nuestro garaje tantos vehículos como queramos, donde podremos seleccionar uno de ellos para realizar nuestra ruta, o borrar cuando queramos alguno de ellos pulsando la “X” que está al lado derecho de cada vehículo.

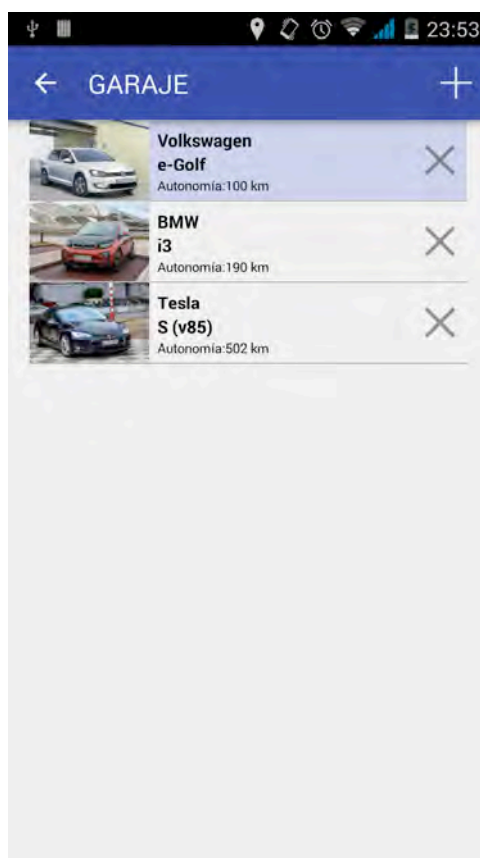


Figura 47: Sección Garaje con un vehículo seleccionado

Al haber seleccionado un vehículo ya dispondremos de sus datos en la pantalla principal para calcular la ruta y su consumo en ella.

Introducimos un destino en el cuadro y aceptamos el destino con el icono que se muestra a la izquierda del cuadro de texto o en el teclado representados con una lupa, obteniendo los siguientes datos:

- Vehículo seleccionado
- Distancia en kilómetros de la ruta seleccionada
- Tiempo de duración de la ruta seleccionada
- Gasto de batería en % que se gastaría al realizar la ruta hasta su destino. Este valor puede tomar varios colores según la batería gastada, siendo verde un gasto hasta el 60%, amarillo entre 60% y 90% y rojo para valores mayores que 90%. Se supone que el coche estará cargado totalmente antes de iniciar la ruta, por lo que si llega al 100% significará que ha gastado toda su batería.

- Marcadores:



Marcador de inicio de ruta.



Marcador de final de ruta.



Marcador de posición actual del usuario.



Marcador de posición donde el vehículo se quedará sin batería. Este dato es aproximado ya que depende de muchos factores que no se pueden predecir como por ejemplo el tipo de conducción de cada conductor.



Marcador de punto de recarga Movele.

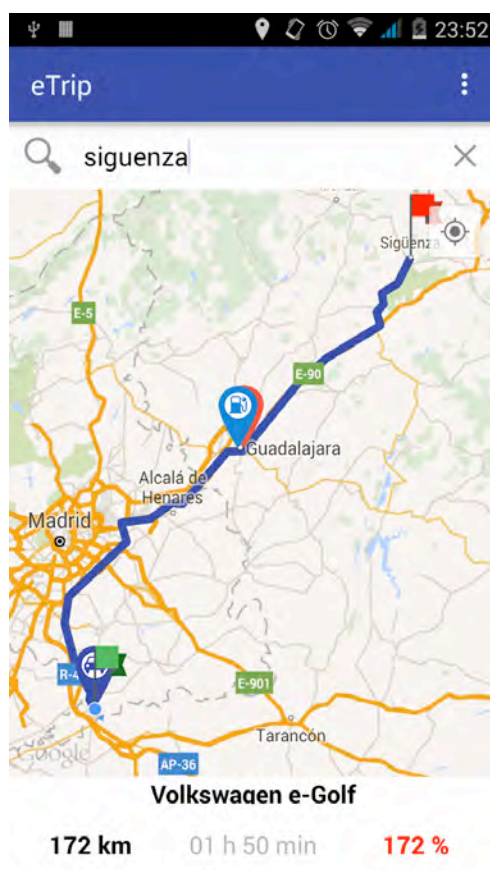


Figura 48: Pantalla principal con una ruta indicada

En la ruta seleccionada podemos observar como con ese vehículo no llegamos al destino, mostrándonos el punto donde nos quedaremos sin batería y donde podemos recargarla.



Figura 49: Puntos de gasto de batería y punto de recarga cercano

La aplicación nos mostrará los puntos de recarga del municipio donde nos vamos a quedar sin batería. Es posible que en algunos municipios nos indique que no existen puntos de recarga o que los existentes estén muy lejos de nuestro punto, esto en un futuro se corregirá al ampliar el plan Movele 2015 donde se implantarán muchos más puntos de recarga.

En el menú superior derecho, disponemos de varias opciones las cuales vamos a explicar:



Figura 50: Menú de la aplicación

- Opciones. Podemos elegir el tipo de ruta a realizar, evitando autopistas, peajes o ferris en nuestro camino.

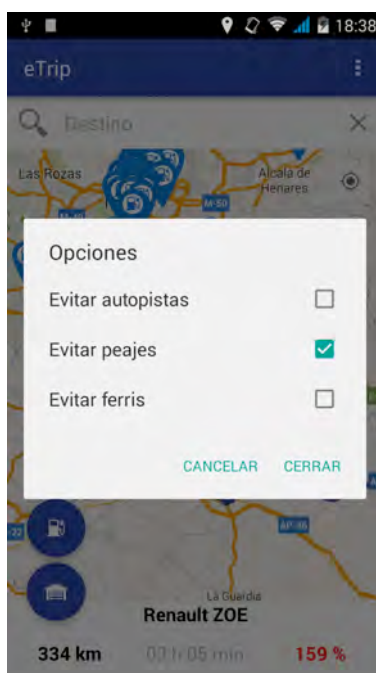


Figura 51: Opciones de la ruta

- Acerca de... Nos muestra información sobre la aplicación, además de incluir los enlaces a los sitios web de la Universidad Carlos III de Madrid y del plan Movele del Ministerio.



Figura 52: Sección Acerca de...

- Cerrar sesión. Nos permite cerrar la sesión del usuario actual para poder iniciar con otro distinto.

Se dispone de un segundo botón flotante para encontrar los puntos de recarga del municipio en el que el usuario se encuentra en ese momento.

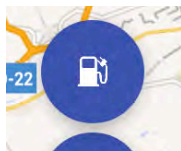


Figura 53: Botón puntos de recarga

Al pulsar ese botón, el mapa se centrará mostrándonos todos los puntos de recarga disponibles, tengamos una ruta seleccionada o no.

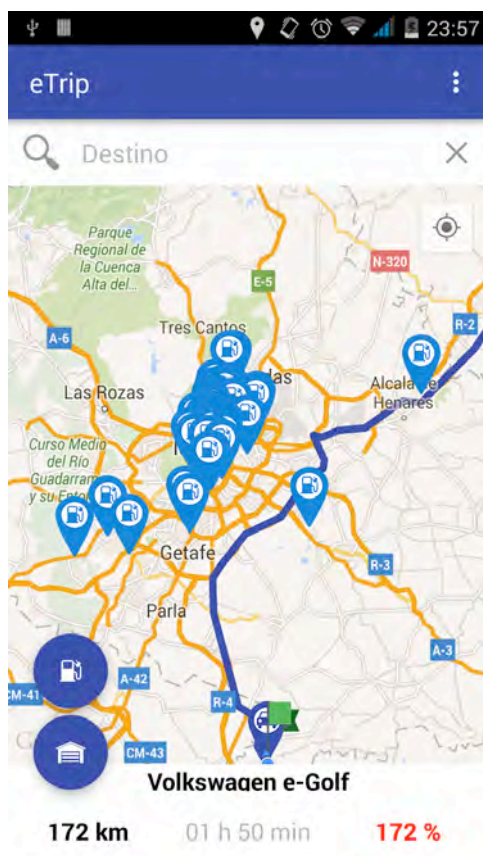


Figura 54: Puntos de recarga de la posición del usuario



Si pulsamos cualquier marcador del mapa, nos dará una pequeña información de él y además aparecerán varias opciones para vincular la aplicación con Google Maps. Estas opciones son:

- Empezar ruta guiada, la cual abrirá la aplicación de Google Maps de nuestro dispositivo para poder empezar la ruta hasta la posición del marcador pulsado.
- Ver el punto del marcador en la aplicación de Google Maps.



Figura 55: Opciones de Google Maps

Si queremos salir de la aplicación debemos pulsar el botón de retroceso de nuestro teléfono móvil, ya que si dejamos la aplicación en segundo plano estará todo el tiempo usando nuestra ubicación para mostrarla en el mapa. Además, si disponemos de una ruta seleccionada y decidimos cerrar la aplicación, se nos mostrará un mensaje para saber si estamos seguros o no de querer cerrarla ya que se perderán los datos de la ruta.



## Capítulo 6. Planificación del trabajo y presupuesto

En este capítulo se presenta la planificación del proyecto realizado, así como el presupuesto necesario para llevarlo a cabo.

### 6.1. Planificación del trabajo

La planificación a la hora de embarcarse en un proyecto es una parte clave en cualquier desarrollo, donde se enumeran las posibles tareas a desarrollar y el tiempo que se va a destinar en ellas. Es la parte del trabajo necesaria para garantizar al cliente una fecha máxima en la cual el proyecto se deberá entregar. Además es una gran herramienta para que los miembros del proyecto se organicen y sepan si se están cumpliendo los objetivos respecto a los plazos fijados.

En primer lugar se deben definir las tareas que se llevarán a cabo estimando el tiempo que se va a invertir en su desarrollo.

1. **Planteamiento del problema.** Se elabora un estudio de los sistemas que contemplarán el proyecto.
2. **Análisis de los sistemas.** Engloba las necesidades y restricciones del proyecto para saber cuánto tiempo se va a emplear en él.
3. **Desarrollo de la aplicación.** Se lleva a cabo la implementación del código de la aplicación.
  - 3.1. Implementación del mapa y rutas
  - 3.2. Inicio de sesión y registro
  - 3.3. Consumo de base de datos
  - 3.4. Punto de gasto de batería
  - 3.5. Diseño interfaz
  - 3.6. Servicio puntos de recarga
  - 3.7. Menú
  - 3.8. Corrección de errores
4. **Pruebas.** Se realizan las comprobaciones necesarias para el buen funcionamiento de la aplicación y el cumplimiento de todos los requisitos establecidos.
5. **Elaboración de la documentación.** Se redacta la documentación técnica con el proceso del proyecto realizado.

Con las tareas del proyecto ya definidas se representan dichas tareas temporalmente, estimando su duración. Se debe de tener en cuenta que la fecha máxima de entrega del trabajo es el 27 de Septiembre, por ello se realizan las estimaciones teniendo en cuenta este plazo.





A continuación se ofrece un diagrama de Gantt, donde podemos ver las tareas y su estimación en el tiempo. Existen semanas donde no se planificó ninguna tarea, ya que por estar trabajando y estudiando a la vez el alumno tuvo que dejar durante un tiempo el desarrollo del proyecto.

	OCTUBRE				NOVIEMBRE				DICIEMBRE				ENERO				FEBRERO				MARZO			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
<b>FASE 1</b>																								
<b>1. Planteamiento del problema</b>																								
<b>2. Análisis del sistema</b>																								
<b>FASE 2</b>																								
<b>3.1. Implementación del mapa y rutas</b>																								
<b>3.2. Inicio de sesión y registro</b>																								

Tabla 51: Diagrama de Gantt (I)

	ABRIL				MAYO				JUNIO				JULIO				AGOSTO				SEPTIEMBRE			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
<b>FASE 2</b>																								
<b>3.1. Implementación del mapa y rutas</b>																								
<b>3.2. Inicio de sesión y registro</b>																								
<b>3.3. Consumo base de datos</b>																								
<b>3.4. Punto de gasto de batería</b>																								
<b>3.5. Diseño interfaz</b>																								
<b>3.6. Servicio puntos de recarga</b>																								
<b>3.7. Menú</b>																								
<b>3.8. Corrección errores</b>																								

FASE 3																				
4. Pruebas																				
5. Elaboración de la documentación																				

Tabla 52: Diagrama de Gantt (II)

La mayoría de las tareas desarrolladas conllevaron más tiempo de lo que estaba previsto. En el diagrama podemos observar varios colores:

- Gris. Planificación inicial
- Verde. Disminución del tiempo previsto, donde la tarea realizada ha sido realizada antes de tiempo.
- Rojo. Incremento del tiempo previsto, por encontrarse dificultades en el desarrollo de la tarea.

## 6.2. Presupuesto

Para el cálculo del presupuesto se han teniendo en cuenta los costes materiales, costes del personal y los costes indirectos en la elaboración del proyecto. Además se añaden los márgenes de beneficio y de riesgo a tener en cuenta.

A continuación se van a describir los costes materiales que se han necesitado para el desarrollo del trabajo:

- **Lugar de trabajo.** Se ha decidido optar por un espacio de trabajo de uso compartido en el centro de Madrid, ya que para este proyecto sólo se ha necesitado el trabajo de un desarrollador siendo este tipo de lugar suficiente para ello. Su alquiler supone un gasto de 125 euros al mes que durante los 10 meses invertidos en el proyecto, sin contar la primera semana de Octubre ni el mes de Enero que no se necesitó este servicio, hace un gasto de 1250 euros.
- **Ordenador.** Ordenador portátil personal Macbook Pro, utilizado para el desarrollo de la aplicación y la posterior documentación, con un coste de 1.499 euros.
- **Dispositivos móviles.** BQ Aquaris E5 HD con un coste de 149 euros, Motorola XT1068 al precio de 149 euros, y Samsung Galaxy Ace 2 (i8160) al coste de 115'99 euros.
- **Software.** Android Studio gratis. Microsoft Word Mac 2011 sin ningún coste al estar dentro del plan Office 365 Ámbito Educativo obtenido gracias al ser todavía estudiante. Adobe Photoshop CC con un coste de 196'02 euros por un abono de un año de duración, siendo un precio especial para estudiantes.

Para estimar el coste del personal se calcula el número de horas empleadas en el trabajo. Para ello consultamos los honorarios profesionales según el COITT, Colegio de Ingenieros Técnicos de Telecomunicación:

“... siguiendo directivas europeas, se debían eliminar los baremos orientativos de honorarios que tradicionalmente veníamos publicando. Por ello, y haciendo notar que los honorarios son libres y responden al libre acuerdo entre el profesional y su cliente...” [12]

Se fija un precio por hora de trabajo de 25 euros. Cada semana se invierten 21 horas, menos las últimas 5 semanas que aumenta el número a 42 horas. Así pues, las horas de trabajo invertidas en el todo el proceso de la realización del proyecto son 882 horas. Al coste del desarrollador se le añade el 21 % de IVA, ya que los demás productos a contabilizar ya disponen del IVA incluido en el precio.

También se deben contabilizar los costes indirectos del proyecto, siendo los gastos ocasionados por la elaboración del proyecto componiéndose de los gastos de luz, teléfono, conexión a internet y material de oficina.

- **Luz.** El espacio de trabajo tiene un gasto adicional de 20 euros al mes por uso compartido de la red eléctrica.
- **Teléfono.** Se ha utilizado la cuenta telefónica propia del desarrollador con un coste de 24'76 euros al mes.
- **Conexión a Internet.** El espacio de trabajo de uso compartido añade un gasto adicional de 15 euros al mes para el uso de internet en sus instalaciones.
- **Material de oficina.** Se ha estimado un coste de material de oficina de 35 euros.

DESCRIPCIÓN	COSTE
Lugar de trabajo	1.250 €
Ordenador	1.499 €
Dispositivos móviles	413'99 €
Software	192'02 €
Costes indirectos	632'6 €
Francisco Vara de la Cruz – Doblado	22.050 € +21% IVA = 26.680'5 €
<b>COSTE TOTAL</b>	<b>30.668'11 €</b>

Tabla 53: Coste del proyecto

A continuación se presenta el beneficio a obtener tras la realización de los costes. Se añade un margen de beneficio del 10% sobre el coste total, siendo tan bajo al ser una aplicación con futuras mejoras con las que se espera obtener un mayor beneficio.

COSTE TOTAL	MARGEN DE BENEFICIO	BENEFICIO TOTAL
30.668'11 €	10%	3.066'81 €

Tabla 54: Margen de beneficio

Adicionalmente, se añade un margen de riesgo sobre el coste total, sin incluir los beneficios, fijado en un 5%.

COSTE TOTAL	MARGEN DE RIESGO	BENEFICIO TOTAL
30.668'11 €	5%	1.533'40 €

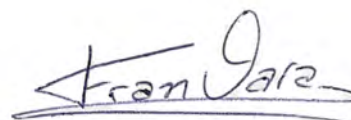
Tabla 55: Margen de riesgo

La siguiente tabla presenta la cantidad total a cobrar al cliente:

DESCRIPCIÓN	COSTE
Coste total	30.668'11 €
Beneficio	3.066'81 €
Margen	1.533'40 €
<b>CANTIDAD TOTAL (IVA incluido)</b>	<b>35.268'32 €</b>

Tabla 56: Cantidad total

La oferta por la realización de este proyecto asciende a un coste de: TREINTA Y CINCO MIL DOSCIENTOS SESENTA Y OCHO EUROS CON TREINTA Y DOS CÉNTIMOS DE EURO.



Fdo. Francisco Vara de la Cruz – Doblado

Graduado en Ingeniería de Sistemas Audiovisuales



## Capítulo 7. Conclusiones y líneas futuras

En el presente capítulo se exponen las conclusiones derivadas de la elaboración de este proyecto. Además, se explicarán las posibles mejoras para futuras actualizaciones de la aplicación.

### 7.1. Conclusiones

A lo largo de esta memoria se ha descrito el desarrollo completo de una aplicación para el sistema operativo Android. Debido a este procedimiento elegido para la realización del proyecto, la aplicación obtenida es estable, ya que se han realizado pruebas en todo momento del desarrollo.

Durante la realización de este trabajo, se han obtenido conocimientos y afianzado los que, gracias a la asignatura Aplicaciones Móviles cursada en la carrera, hicieron al alumno adentrarse en esta tecnología.

Cabe destacar el hecho de haber sido capaz de desarrollar una aplicación desde cero. Este hecho no había sido realizado durante la carrera, ya sea porque se mejoraban aplicaciones ya empezadas aportadas por los profesores o por la realización en grupo de las mismas, donde a cada alumno se le asignaba una parte del proyecto. Por ello, los objetivos del presente proyecto han sido cumplidos en su totalidad, pretendiendo dar continuidad en el futuro con las mejoras que se exponen a continuación.

### 7.2. Líneas futuras

A continuación se proponen algunas mejoras para realizar en un futuro en la aplicación.

Un punto a mejorar en la aplicación es dar al usuario más puntos de recarga para que encuentre uno cercano al punto donde se quede sin batería. Ahora mismo la aplicación sólo indica los puntos proporcionados de forma gratuita por el plan Movele, siendo pocos en algunas comunidades. Una opción para dar un mejor servicio es crear una base de datos propia donde se tengan los puntos de recarga que están fuera del plan del Ministerio, como son los instalados en parkings, aeropuertos, en plena calle, en centros comercial y en estaciones de servicio, entre otros muchos lugares.



Además de proporcionar una lista al usuario, se le daría la oportunidad de enviar la posición de los puntos que se encontrara por el camino, para así los responsables de la aplicación pudieran incorporarlo a la base de datos, cerciorándose de que existen. Estos nuevos puntos se indicarían con otro icono, facilitando información sobre él como precios y estado de las instalaciones.

Otra posible mejora sería perfeccionar el cálculo del consumo de batería. Para ello se utilizaría Roads API, un API de Google que indica la velocidad máxima de cada carretera, para así realizar cálculos más exactos. Este servicio no se ha utilizado para el proyecto al ser una herramienta de pago. Se recogerían más especificaciones del vehículo de su ficha técnica para los cálculos, además de pedir al usuario información de pasajeros en la ruta para poder añadir al peso del vehículo el peso adicional medio por persona.

Como último perfeccionamiento de la aplicación, siendo el más importante, se contactaría con alguna marca de vehículos eléctricos para que, con su tecnología, se diseñara un tipo de conexión segura a la aplicación para obtener los datos de autonomía real del vehículo. Con ello se podría calcular de una manera más real donde se quedaría sin batería el usuario y así, kilómetros antes, poder buscar en la ruta un punto de recarga válido para el tipo de vehículo.



## Bibliografía

- [1] Professional Android 4 Application Development. Reto Meier. John Wiley & Sons, Inc. 2012 ISBN 978-1-118-10227-5
- [2] Global Stats. Stat Counter:  
<http://gs.statcounter.com/>
- [3] Google. Información sobre la empresa:  
[https://www.google.com/intl/es\\_es/about/company/](https://www.google.com/intl/es_es/about/company/)
- [4] Baas. Los servicios llegan al backend:  
<http://www.computerworld.es/cloud/baas-los-servicios-llegan-al-backend>
- [5] Diccionario de la Real Academia Española:  
<http://www.rae.es/recursos/diccionarios/drae>
- [6] Diario Motor. La historia del coche eléctrico:  
<http://www.diariomotor.com/tecmovia/2011/07/19/la-historia-del-coche-electrico-un-largo-proceso-de-desarrollo/>
- [7] Proyecto de Movilidad Eléctrica Movele:  
<http://www.movele.es/>
- [8] Boletín Oficial del Estado. Programa MOVELE 2015:  
<http://www.boe.es/boe/dias/2015/04/18/pdfs/BOE-A-2015-4215.pdf>
- [9] Draw Use Case Diagram:  
<http://yuml.me/diagram/scruffy/usecase/draw>
- [10] Google Maps APIs:  
<https://developers.google.com/maps/documentation/directions/intro>
- [11] Pro Git book, Scott Chacon and Ben Straub:  
<https://git-scm.com/book/es/v2>
- [12] COITT, Honorarios profesionales  
<http://www.coitt.es/res/libredocs/Honorarios.pdf>



## Chapter 1: Introduction

### 1.1. Motivation

Vehicles are a required mean of transport in the current lifestyles, but it is one of the main sources of pollution in the cities. Traffic pollution is caused due to the burning of fossil fuels such as diesel oil or petrol originating from internal combustion vehicles, which are the most used among drivers.

Emissions coming from vehicles in circulation cause an enormous environmental problem. These emissions contribute to the greenhouse effect and the depletion of non-renewable materials. To solve these difficulties, new concepts of mean of transport appear.

European Commission has proposed a Global Plan to eradicate the traditional vehicle in 2050. There are lots of plans operating throughout the world that are unknown.

On the other hand, mobile devices have become a fundamental tool for our social and working lives because of the fact that these devices have immensely evolved.

These facts mean a huge motivation to accomplish this project. Apart from extending author's knowledge on the field, the fact that technology may help to eradicate the lack of awareness about environmental problems entails an extra motivation.





## 1.2. Objectives

In the present TFG, there are two main objectives. Firstly, the development of an Android application directed to maps. This app will provide users with a positive experience with useful information and easy management. Its graphic interface must be simple as well as trendy according to the new tendencies. Both aspects are not usually carried out by developers on their own if there is a designing team that guides them on the process. Therefore, during the process of this project both developing and designing tasks have been accomplished in order to the app to be simple and trendy.

Secondly, another objective will be the creation of a project which contributes to users' awareness of electric vehicles. It will present aid plans Movele by Minister and prove a rising autonomy of electric vehicles when performing a route.

This app will be able to implement the functionalities below:

- Management of Google Maps
- Autonomy estimation from a suggested route
- Charging Stations for electric cars when its battery is completely consumed
- Nearest charging stations to users' location

## 1.3. Structure of the report

The current report of the TFG is itemised in seven sections described below:

- Section 1. Compile the motivation to accomplish the project as well as the objectives and the structure of the report.
- Section 2. Present and analyse the necessity to fulfil this project, determining techniques and technologies that have been taken into account to design the process.
- Section 3. It makes reference to the requirements of capacities, functionalities and restrictions of the systems. It also includes possible alternatives.
- Section 4. Propose technical solution and explain the method to perform the project.
- Section 5. Compile the auditions carried out in order to reach the final version. Moreover, a manual for users with the available options of the app is supplied.
- Section 6. Present the planning and the estimate costs for the fulfilment of the project.
- Section 7. Expound the final conclusions and lines for future research to improve the app.



## Chapter 7: Conclusions and future lines of research

The current section set the conclusions deduced from the creation of this project. Furthermore, possible improvements for future updates of the app are explained.

### 7.1. Conclusions

Throughout this report, the full development of an app has been thoughtfully described and tested. Due to this process, the resulting app is highly stable.

Thanks to the subject Mobile Applications which introduced the author into this technology, during the accomplishment of the project, author's knowledge on the field has been reinforced and extended.

It is remarkable the fact of being able to create an app from the beginning. In the degree, students were required to improve applications or to accomplish a section of the process. For that reason, main objectives have been totally achieved. Additionally, it is expected to carry on future improvements presented below.

### 7.2. Future lines of research

According to this study, one of the aspects which may be improved in the future would be the creation of a database with all the charging stations apart from those included in the Movele plan. This necessity emerges from the fact that the list of those charging stations near to the location where the vehicle runs out of battery is not enough. It is even worse in certain communities.

This new database would include charging stations such as those installed in car parks, airports, shopping centres, service stations and those along streets that are excluded from Movele plan.

In addition to this function, this service would also offer the chance to introduce those charging stations which users may find on their way and are not included on the list. In case that its existence is checked, this new charging station would be enlisted and added extra useful information such as price and the condition of the station.



Another possible improvement would be the execution of more accurate estimation of the battery consumption. In order to deal with this improvement, Roads API –an API by Google which points out the speed limit in every road- would be used. This service has not been included in the project because it is not for free. It would be compiled vehicle specifications from its technical datasheet to perform the estimation. Furthermore, the users would be asked for providing information about the number of passengers in order to add the additional average weight per person to the vehicle weight.

Finally, the most important improvement to take into consideration would involve being in contact with electric vehicles companies in order to create a safe connection to obtain accurate data about the real autonomy of the car. Thanks to their technology, users would be aware of the kilometres before the next valid charging station to their kind of vehicle.



## SUMMARY

Vehicles are a necessary mean of transport for our present-day society, and also one of the largest sources of pollution in our cities. This is because of the fact that most vehicles in circulation have an internal combustion engine. Nowadays, new transport concepts are showing up in order to reduce the environmental problem and in order not to contribute to the greenhouse effect. New alternatives are hybrid vehicles, which have already been quite implanted in society and electric vehicles, which are one of the best alternatives due to progresses in new technologies.

Electric vehicles contribute with a lot of benefits in the field of energy efficiency such as: reduction of the use of petroleum products, reduction of CO<sub>2</sub> emissions and, of course, other pollutants emissions related with the greenhouse effect and the non-deterioration of the greenhouse effect itself. Because of what has been previously said, this kind of vehicle helps us in order to improve our air quality and decrease noise pollution, an increasing problem in big cities.

An electric vehicle can be defined as a mean of transport that works by means of electricity. According to this definition, this kind of vehicle comprehends a wide range of means of transport, not only differentiated by the form or the environment where they travel, but also by the way in which the electric energy is obtained, for example:

- Electric trains that use an external source of energy.
- Solar vehicles, which use renewable energy.
- Ships which generate their energy on board.

In this project we will focus on vehicles that use the electricity to propel themselves, since this kind of vehicle are the closest to the user.

The electric motor has possessed since its invention some advantages, which until today remains the same:

- Little noise pollution.
- Little air pollution.
- Robustness.
- Does not use a limited raw material, such as petroleum.
- Much lower maintenance and cost.
- Easily usable.



Despite these advantages, it also possesses some other technological limitations:

- Too much time in order to charge the batteries.
- Limited outputs.
- Little autonomy.
- Some of these vehicles were still polluting, since they used the electricity which was obtained from contaminant raw materials.

With these limitations, this type of vehicle was not very much used in the past, and it was vehicles with combustion engines which were more successful. At present times, these disadvantages have been improved but are not known or little known by users.

The Ministry of Industry, Energy and Tourism has developed an Action Plan called “Movele”, within the Comprehensive Strategy for Promoting the Purchase of an Electric Vehicle in Spain (Estrategia Integral de Impulso al Vehículo Eléctrico en España). This strategy is done for promoting the development and the use of electric vehicles in Spain, where is provided the adaptability of the electrical infrastructure for the management and charging them, and also promoting programs of informing and communicating this technology to the general public.

Since the beginning of 2013, this incentive program has contributed to boost the purchase of more than 10,000 electric vehicles in Spain until the end of 2014. This will result in the following data: (throughout the complete working live of the vehicle) the emission of 150,000 CO2 tons will be avoided and 300,000 barrels of petroleum will be saved.

In the same way as it occurs in the PIVE plans, it is provided an equivalent discount to the government subsidy by manufacturers and importers who want to join the Movele 2015 program. In addition to this incentive, (unless that the purchaser refuses to it) a charging station linked to the beneficiary is installed.

Spain Government has provided 761 charging stations for public use, with different types of charging, such as: normal charging, fast charging, motorcycles and vehicles adapted for the disabled people. To gain access to those stations free of charges, it is needed to apply for a card for charging the vehicle, according to each Autonomous Community of Spain. This Movele program card entitles the user to charge the vehicle with electricity for free.



From the end of 2014 to April 2015, there was a period of time when the government support was ended and the new edition program was not already approved. During this time the stations were pretty much forgotten and a bunch of electricity posts were out of order. In the desire that this would not happen again, the management and maintenance of these stations have been given to some companies, which started the restructuring and updating works of the charging points that were functioning, and moreover, new ones were installed. It is expected the technical and usage conditions will change and it is possible that in a near future charging the vehicles will no longer be for free, but maintaining a discount on it.

In the website of the Movele program (<http://www.movele.es/>) it can be located the available charging stations in Spain within the program.

In the present final project, two main objectives are being pursued. The first objective would be the creation of a project which contributes to the electric vehicles awareness for the general public. It will show up the Movele aid plan (made by the Government) and will show the growing autonomy that this type of vehicle has achieved in order to perform a trip.

The second objective would be the development of an Android app oriented for maps. The knowledge, which has been obtained during the degree, has been used as an introduction to this new matter in which this project is completely plunged into. Because of this fact, it was decided to choose a project which goes into detail about the subject, in order to understand the complexities of its operational process.

These facts mean a huge motivation to fulfill this project. Since, not only give the student the opportunity to improve his abilities in this particular field, but also this project brings into the light the possibility that the technology may help stopping the ignorance about all these action plans, which may alleviate the huge environmental problems that our society is facing.

The Android operating system was originally designed for mobile devices such as smartphones and tablets, although when it was bought by Google in 2005, it was modified in a way that it can be used in a wide range of devices.

Android is based on a modified version of the Linux kernel which turns the Java app code into a code that is easily understood by the core, through its runtime environment. It is an open source system, so that anyone can create and develop apps for this system.



The Android app which was made for this Final Degree Project (Trabajo de Fin de Grado) implements the following functionalities:

- Management of Google Maps.
- Calculation of autonomy according to the specified route.
- Displaying charging stations for electric vehicles nearby the point where the user would consume all his/her vehicle battery.
- Indicate the charging station in the nearby of the current location of the user.

The app presents a simple graphic interface, according to the nowadays trends. These aspects are usually carried out by a design group, who advise the developers, so they do not have to do the design part on their own. For this particular project, the developer must perform both develop and design tasks.

To obtain the necessary results, the application gives us the option to choose among a wide range of electric vehicles available in the commercial market. For this, the app uses a database from a cloud service. A Baas service was chosen (Backend as a Service) which is offered by the company Parse (<https://www.parse.com/>) because of the fact that it was free, because of the services that are used by the app, and its simplicity in use compared to other companies thanks to its SDK (Software Development Kit).

The app examines the background database every time you want to change the selected vehicle, obtaining the necessary data in order to calculate the autonomy of the vehicle during the route which was indicated by the user.

The database is also used to identify each user with his/her login data. This action enables us to know which vehicle is owned by which user, and the data are kept in a personal garage, where the vehicle about to be used can be changed.

Internet is required to use the app, as the following functionalities make use of it:

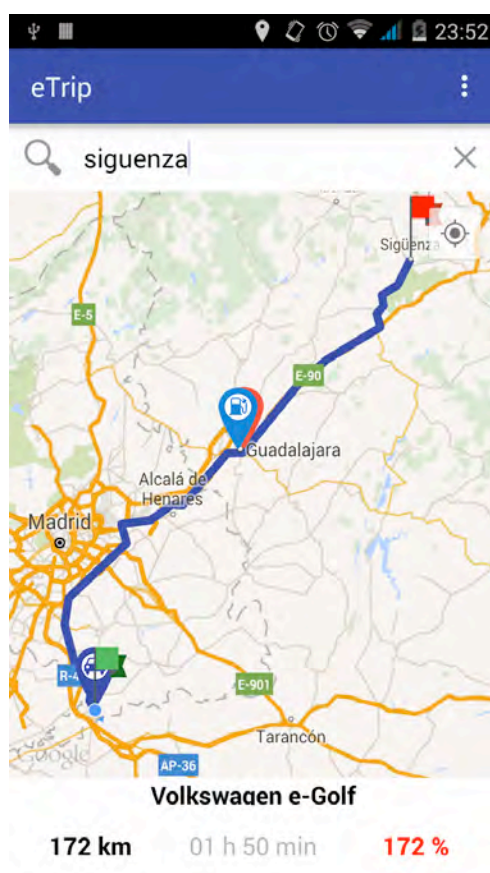
- Downloading maps.
- Calculating the route indicated by the user.
- Choosing the vehicle from database.
- Displaying the electric vehicles charging stations.

In order to show the electric vehicles charging stations, they were used the ones given by the Government Movele Plan. The Plan Movele web page does not dispose of such a service which can be used by our app. Because of this, it was decided to parse this web, looking for the essential data about the charging stations.

As a result of this parsing, the needed data were kept, so we could add to our map the marking points of the charging stations. These points are shown to the user in two different occasions:

- Showing the electric vehicles charging stations nearby the point where all the battery would be consumed.
- Indicating the charging stations nearby the current location of the user.

When the vehicle autonomy is not enough to reach the selected destination point, the app shows an approximate point where it would be run out of battery. The app contains the town name of that point and add the electric vehicles charging stations available in that town or nearby.



Picture 56: Main screen





In this Final Degree Project it is described the whole development of the app for the Android operative system. Due to the selected procedure in order to develop this project, the obtained app is stable, since many tests have been fulfilled during all the development process.

During the accomplish process of this task, many and important knowledges have been obtained, moreover, the knowledge obtained in the Mobile Apps course (Asignatura de Aplicaciones Móviles) which was done during the degree has been consolidated and thanks to this course, the student had the possibility of entering in such a technology.

I would like to highlight that fact of being able to develop a whole app from zero. This fact has not been previously performed during the degree (either because we continued an app, which was already started by a teacher or we did it in groups, where each student had to accomplish a part of the project).

Because of this fact, the objectives of the present project has been totally accomplished, with the expectation of being able to continue them in the future, with the improvements which are exposed hereafter, during the written part.